AFRL-SR-AR-TR-09-0119

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*
28-02-2009

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
May 2006 – November 2008

4. TITLE AND SUBTITLE
Optimal Robust Matching of Engine Models to Test Data

5a. CONTRACT NUMBER
FA9550-06-1-0299

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Mavris, Dimitri N.

Denney, Russell K.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Georgia Institute of Technology
School of Aerospace Engineering
Atlanta, GA 30332-0150

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NA
801 Randolph St.
Arlington, VA 22202-1977

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Status matching supports USAF turbine engine development, qualification, and maintenance test planning and diagnostics. Future USAF maintenance concepts will require that engine status decks be made more frequently than they are today, and thus the process must be faster and require less expert knowledge than the traditional approach.

The research program developed an improved, automated process for calibrating turbine engine performance models. The Filtered Monte Carlo (FMC) and the Singular Value Decomposition (SVD) algorithms were found to meet the sponsor's requirements for a robust, fast process suitable for inexperienced users. Both methods were demonstrated to successfully match measured data with no prior knowledge of the engine. The methods are complementary in that an initial FMC analysis can identify for an inexperienced user which model tuning parameters are significant and can provide him or her with appropriate ranges for those parameters. The SVD method may subsequently be used to quickly determine the best value for each modifier.

The engine status matching process is also applicable to calibration of other types of models. Similar methods have been used by the researchers for calibration of engine, aircraft, noise, and emissions models and for calibration of lower fidelity aero models to CFD models.

15. SUBJECT TERMS
Gas turbine engine; Performance model; status matching

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Russell K. Denney
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	58	**19b. TELEPHONE NUMBER** *(include area code)* (404)385-6214

20090429210

Optimal Robust Matching of Engine Models to Test Data

A Final Report to
Air Force Office of Scientific Research
Arlington, VA

AFOSR Grant No. FA9550-06-1-0299

Period of Performance:
01 May 2006 – 30 November 2008

Submitted by:

Dr. Dimitri Mavris, Principal Investigator
Mr. Russell Denney, Co-Principal Investigator

Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA
www.asdl.gatech.edu

28 February 2009

Executive Summary

Status matching supports USAF development, qualification, and maintenance test planning and diagnostics. Thus improved status matching is a key enabler for improved testing and maintenance processes within the USAF. Improved methods may also support future USAF maintenance concepts such as providing customized status models for each engine in the fleet. Under such scenarios, status decks will need to be made more frequently than they are today, and thus the process must be faster and require less expert knowledge than the traditional approach.

This research program developed an improved, automated process for calibrating turbine engine performance models. The research proceeded in three phases: (1) a literature search for algorithms applicable to the status matching problem, (2) a preliminary investigation using simulated engine data of several of the methods identified in the literature search, culminating in the selection of two methods, the Filtered Monte Carlo (FMC) and the Singular Value Decomposition (SVD) methods, for further investigation, and (3) the detailed investigation of the selected algorithms using real engine data.

The proposed algorithms were found to meet the sponsor's requirements for a robust, fast process suitable for inexperienced users. Both methods were demonstrated to successfully match measured data with no prior knowledge of the engine. The methods are complementary in that an initial FMC analysis can identify for an inexperienced user which variables are significant and can provide him or her with appropriate ranges for those variables. The SVD method may subsequently be used to quickly determine the best value for each modifier.

One of the major conclusions of the research is that the choice of solution algorithm is not the most significant issue. All the algorithms investigated are theoretically similar to each other. The most significant factor in a successful method is the user's choice of modifiers. It is difficult to avoid the fact that this requires experience; although certain steps in the process have been automated and effectively prompt the user when a decision is required. It is noteworthy that, if user experience can be translated into probability distributions for priors, then Bayesian methods can easily be incorporated into the developed process. Over the long term, this may prove to be the best approach to the problem.

Finally, it should be noted that the engine status matching process is applicable to calibration of other types of models. Similar methods have been used by the researchers for calibration of engine, aircraft, noise, and emissions models and for calibration of lower fidelity aero models to CFD models.

Table of Contents

Executive Summary i
Table of Contents ii
List of Figures iv
List of Tables v
1 Problem Statement / Objective 1
2 Background 1
2.1 Status Deck Description 1
2.2 Characteristics of the Problem 4
3 Literature Review 7
3.1 AGARD Publications 8
3.2 Engine Diagnostics Literature 8
3.3 Optimization Methods 9
3.4 Surrogate Modeling 10
3.5 Other Statistical Methods 10
3.6 Hydrological modeling literature 11
4 Algorithm Selection 11
4.1 Sample Problem 12
4.2 Algorithm Selection Results and Conclusions 14
5 Description of Selected Algorithms 19
5.1 Filtered Monte Carlo (FMC) Method 19
5.2 Singular Value Decomposition (SVD) 20
5.3 Measurement uncertainty considerations 23
6 Demonstration 23
6.1 Description of the Data 23
6.2 Baseline Engine Model 24
6.3 User Interface 26
6.3.1 Interfacing Measurements 27
6.3.2 Interfacing independent variables 28
6.4 Results of SVD Method 29
6.4.1 Representative Engine (SN# 716310) 29
6.4.2 Entire Engine Population 35
6.5 Results of FMC method 35
6.5.1 Histograms 36
6.5.2 Correlations 38
6.5.3 Principal Component Analysis 42
6.5.4 Fisher's Linear Discriminant Analysis (LDA) 44
6.5.5 Additional Monte Carlo Studies 45
7 Conclusions and Recommendations 47
7.1 Impact / Review of Significant Results 47
7.2 Transition / Collaboration Opportunities 47
7.3 Recommendations for Future Work 48
7.3.1 Modifier selection 48
7.3.2 Regression 48
8 Acknowledgements 49

9 References 49

List of Figures

Figure 1: Components of a typical turbojet engine cycle deck 2
Figure 2: Typical balance equations for a turbojet engine cycle deck 3
Figure 3: An Example of Unstable Inverse Solutions [40] 7
Figure 4: Engine Model and Nomenclature for Sample Engine Model 13
Figure 5: Cycle Model vs. RSE Execution Time 15
Figure 6: Schematic of Filtered Monte Carlo process 19
Figure 7: Flowchart of SVD Calculations 22
Figure 8: Schematic Diagram of NPSS Engine Model Components 24
Figure 9: PW2037 Power Hook 25
Figure 10: Baseline Model Comparison with Engine Test Data 26
Figure 11: User Interface 27
Figure 12: Percent Error for Engine SN# 716310 Band A (100% Power) 30
Figure 13: Modifier curve fits for Fan and LPC 32
Figure 14: Modifier curve fits for Splitter and HPC 33
Figure 15: Modifier curve fits for HPT and LPT 34
Figure 16: Representative Plots of Status Deck vs. Baseline and Test Data 34
Figure 17: Non-uniform Histograms of Performance Modifiers 38
Figure 18: Correlation Plots 42
Figure 19: Projected Data onto the Principal Component Space 44
Figure 20: Contribution Plot 45
Figure 21: Number of Points Remaining for a Given Filter Tolerance 46
Figure 22: Monte Carlo Results Independence 46

List of Tables

Table 1: Simulated Measurements for MFTF 13
Table 2: Simulated Performance Modifiers for MFTF 14
Table 3: Converged Results and Closeness to Measurement Data: EKF and UKF 15
Table 4: Converged Performance Modifier Values for EKF and UKF 16
Table 5: Algorithm Advantages and Disadvantages 18
Table 6: PW2037 Measurements 28
Table 7: Component Performance Modifiers 29
Table 8: SVD Results for Engine SN# 716310 30
Table 9: Final Values for Status Deck – Engine SN# 716310 31
Table 10: Ranges of the Engine Performance Modifiers 36
Table 11: Means of the Performance Modifiers 37
Table 12: Matching Results 37
Table 13: Correlation Coefficients 40

1 Problem Statement / Objective

The fundamental motivation for the research reported herein is the need to quickly and accurately match gas turbine engine cycle model predictions to engine test data. The problem is challenging because it requires the solution of an ill-posed, underdetermined nonlinear system. Current solution methods are time consuming and highly dependent upon the experience of the data analyst.

Status matching supports USAF development, qualification, and maintenance test planning and diagnostics. Thus improved status matching is a key enabler for improved testing and maintenance processes within the USAF. Improved methods may also support future USAF maintenance concepts such as providing customized status models for each engine in the fleet. Under such scenarios, status decks will need to be made more frequently than they are today, and thus the process must be faster and require less expert knowledge than the traditional approach.

The objectives of the present research were to investigate a simpler, more automated process that would enable considerable reductions in the time and cost to match engine performance models to measured data, while also improving model accuracy. The approach taken was to survey the methods available to solve the problem, and then to select a small subset of available algorithms for further investigation. The most promising approaches were then applied to a representative engine matching problem and refined into a practical tool available for use in Air Force engine testing and data reduction tasks.

2 Background

2.1 Status Deck Description

A turbine engine cycle model or "cycle deck" is a detailed thermodynamic representation of a turbine engine, comprising semi-analytical, "zero-dimensional" representations of each of the engine components. The term "cycle deck" indicates that the complete thermodynamic cycle is computed and that the operating points of each of the components have been matched so that the engine is in thermodynamic equilibrium. A "status deck" is simply a cycle deck which has been "tuned" or adjusted to match a specified set of measured test data. A status deck may represent a specific engine or an average of a population of engines.

To illustrate, the components of a typical status deck for a single spool turbojet engine are depicted in Figure 1. First, flow computation stations are identified by number at the entrance and exit of each engine component. Each engine component is represented by performance "maps", which are derived through a combination of analytical and empirical means. For example, the compressor and turbine maps are digitized tables which provide pressure ratio and efficiency as a function of corrected rotor speed and corrected flow. For computational purposes the compressor flow, pressure ratio, and efficiency are tabulated along arbitrary ray lines or "R-lines", while the turbine flow and efficiency are tabulated as a function of the turbine pressure ratio and corrected rotor speed. The nozzle map provides the nozzle flow as a function of the nozzle pressure ratio. Pressure drops across the inlet, the combustor, and the exhaust nozzle are also represented as functions of the appropriate parameters.

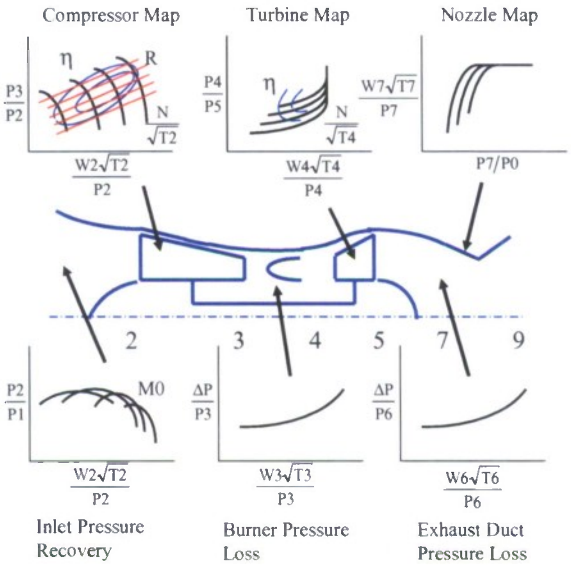


Figure 1: Components of a typical turbojet engine cycle deck

The cycle balance or match point is found by solving a system of simultaneous nonlinear equations using a Newton-Raphson iteration technique (see Figure 2). The equations to be solved represent the conservation laws; continuity and work balances must be satisfied throughout the engine at the match point. The constructions of the maps themselves provide the basis for the computational procedure. For the turbojet example shown, there are five equations: the continuity balances at the compressor entrance, the turbine entrance, and the exhaust nozzle entrance; the work balance between the turbine and the compressor; and the throttle or power setting requirement represented in the example by the turbine inlet temperature demand. Each of these terms is to be driven to zero ($\vec{y}_\infty \rightarrow 0$ within an acceptable tolerance), by varying a set of five independent parameters ($\vec{x}$). In the example shown these independent variables $\vec{x}$ are the compressor corrected rotor speed, the compressor R-line value (which determines the compressor flow, pressure ratio, and efficiency for a given corrected rotor speed), the inlet flow, the turbine pressure ratio, and the combustor fuel-air ratio.

$$\vec{y} = \begin{bmatrix} \left(\frac{W2\sqrt{T2}}{P2}\right)_{MAP} - \left(\frac{W2\sqrt{T2}}{P2}\right)_{CALC} \\ \left(\frac{W4\sqrt{T4}}{P4}\right)_{MAP} - \left(\frac{W4\sqrt{T4}}{P4}\right)_{CALC} \\ \left(\frac{W7\sqrt{T7}}{P7}\right)_{MAP} - \left(\frac{W7\sqrt{T7}}{P7}\right)_{CALC} \\ Q_C - Q_T \\ (T4)_{DEMAND} - (T4)_{CALC} \end{bmatrix} \qquad \vec{x} = \begin{bmatrix} \frac{N}{\sqrt{T2}} \\ R \\ W1 \\ \frac{P4}{P5} \\ FAR \end{bmatrix}$$

$$\vec{x}_i = \vec{x}_{i-1} + \left[\frac{\partial y}{\partial x}\right]^{-1} * (\vec{y}_\infty - \vec{y}_{i-1})$$

Figure 2: Typical balance equations for a turbojet engine cycle deck

It should be noted that this problem formulation is not unique, and the "best" choices for the independent and dependent variables are found through experience or user preference. In any case, once the cycle is balanced, the pressures, temperatures, and flows are known at each of the flow stations, and overall performance values such as thrust and fuel flow may be computed.

During an engine test, pressures and temperatures may be measured at some of the flow stations within the engine, along with other values such as the rotor speeds, the inlet flow rate, thrust and fuel flow. Fundamentally, the process of making a status deck involves adjusting the component performance maps until the computed cycle match point corresponds with the measured parameters. To facilitate these adjustments, the cycle deck is provided with modifiers on the flow, efficiency, and pressure loss values read from each component map.

The process of status matching may be broken down into three steps. First, the test measurements are used to determine the modifiers on the component maps at each data point. Second, the modifiers are regressed to develop curves such that the modifiers may be allowed to vary realistically throughout the engine operating range. This step requires the determination of the functionality of the modifiers; for example, the compressor efficiency modifier may vary as a function of the corrected rotor speed, variable stator vane position, compressor entrance Reynolds number, and compressor rotor-to-casing clearances. Once these functions are determined and the status curves are developed, thc final step in the process is to incorporate these curves into the engine model and to verify the results against the original test data.

The traditional approach to status matching is very much a "hands-on" process. The modifiers must be adjusted manually until the "best" values, in the judgment of the analyst, are found that make the model predictions match the test measurements. This process is highly dependent on the prior knowledge and experience of the analyst. Since it is a manual process, the analyst is usually only able to look at one parameter at a time. Thus the process is very time consuming and computationally expensive. The proposed new approach must be more automated, to require less time and experience on the part of the data analyst. To achieve this goal, the algorithms must be mathematically and computationally stable and reliable.

2.2 Characteristics of the Problem

As described above, the status matching process actually comprises a pair of regression problems. The first problem entails the determination of the model modifiers and the assignment of values to the modifiers at each data point. The second problem entails the determination of appropriate functional relationships to describe how the values of each modifier changes with changing flight conditions and/or engine power settings. Thus the status matching process is plagued with all the issues common to any regression analysis. Typical issues are discussed briefly below:

Causality

Regression analysis cannot by itself establish causal relationships. Goodness-of-fit statistics only reflect the correlation structure of the data being analyzed. Causality can only be determined from controlled experiments. No variable selection procedure can substitute for the judgment of the analyst. In other words, the selection of relevant explanatory variables should be based upon theoretical considerations; empirical methods for variable selection based only on statistical analysis of the test data will tend to be sample specific [13].

Specification errors

Specification errors are errors in identifying the significant explanatory variables. There are two types of specification error. A type I error is "finding something that is not there", e.g. including explanatory variables in the regression which in fact have no significant effect on the observations. A type II error is "missing something that is there", e.g. excluding an explanatory variable which should be in the regression.

Multicollinearity

Multicollinearity, sometimes also called collinearity, refers to the situation when two or more explanatory variables have the same effect on the observations and it is not possible to differentiate between them. This problem is often caused by the limited instrumentation used in full scale engine testing. In most cases, the effect will be "smeared" across the explanatory variables. In extreme cases multicollinearity can cause incorrect signs on the regression coefficients.

Parsimony

Parsimony refers to the historical "Occam's razor" principle of incorporating as few explanatory variables into the model as possible. Originally this principle may have been driven more by computational limitations than by mathematical considerations. However, when the number of explanatory variables is large relative to the number of observations there is a danger of "overfitting" the model. An overfit model fits the noise in the data; i.e. it incorrectly attributes random noise to causal factors.

Random error

The term "random error" is not related to the random noise or measurement error present in experimental data, but refers to the situation when the values of the explanatory variables vary randomly. Classical regression analysis requires that the explanatory variables be set by the experimenter in a designed or controlled experiment. As described above, this is required to prove causality. It is also important to the proper interpretation of goodness-of-fit statistics. For example, the significance of a high R^2 value depends strongly on the range and distribution of the values of the explanatory variables.

It should be clear that the typical status matching problem, due to limited data and limited instrumentation, violates most of the standard regression guidelines. Engineering judgment is often required to resolve the issues which may arise. Thus creating a more automated process which relies less on engineering experience is a daunting task. As will be seen, the approach ultimately selected combines complementary procedures designed to provide guidance to the analyst without taking him or her completely out of the loop.

A more complete understanding of the problem may be obtained by considering the overall process of physical system modeling. Tarantola [34] generalizes a scientific procedure for the study of a physical system with the following three steps:

1. Parameterization of the system
2. Forward modeling
3. Inverse modeling

First, the parameterization of the system is to discover the minimal set of parameters that completely characterizes the system. Such a set is called model parameters. Second, forward modeling is to discover a physical law that allows us to predict the outcome of the system given the model parameters. Third, inverse modeling is to estimate the model parameters when the outcome of the system is observed. According to Tarantola's generalization status matching is an inverse modeling.

Problems that involve the forward and inverse modeling are referred to as direct and inverse problems, respectively. Keller [22] provides a definition of direct and inverse problems in a historical point of view. Keller defines two problems as direct and inverse problems if the formulation of one involves the solution of the other. Among the two problems the direct problem is the one that has been studied extensively than the other while the inverse problem is the one that is less studied or understood than the other. On the other hand, Bertero's definition [2] is based on causal relationships. A direct problem is formulated based on a physical law specifying a cause-effect consequence. The corresponding inverse problem is to find the unknown cause of known effect. Hansen's point of view is more or less similar to that of Bertero. Hansen [17] describes that inverse problems involve finding the internal structure of a system from the observed behavior of the system or determining the unknown input of the system from the known output.

Direct problems are perceived to be much easier than inverse problems, due to the following two properties: locality and causality. Laws of nature are often expressed as a system of algebraic or differential equations. The equations are local in a sense that they express the dependency of the function describing a system and its derivatives on the outcome of the system at a given point, i.e., at a given model parameters. They are causal in a sense that the outcome depends on the model parameters. On the contrary, inverse problems are often not local and/or not causal. Bertero and Boccacci [3] argue that the conceptual difficulty associated with inverse problems due to a loss of information. A forward modeling always involves a loss of information or an increase in entropy. Consequently, an inverse modeling of the same system becomes different from the forward modeling, and the inverse problem requires the recovery of the lost information. The argument of Bertero and Boccacci is an analogy of forward and inverse modeling to an irreversible thermodynamic process.

The conceptual difficulty of inverse problems imposes unfavorable characteristics on inverse problems. According to Hadamard [16] a problem is well-posed if the following conditions are met:

1. a solution exists,
2. the solution is unique, and
3. the solution depends continuously on the data.

Unfortunately, inverse problems are typically ill-posed; one or some of the above conditions are not met for inverse problems. Each unfavorable characteristic of ill-posed problems poses different issues in choosing or implementing a solution technique. Some of these issues are discussed in the following.

First of all, the possible non-existence of inverse solutions makes it hard to set a stopping criterion for solution search. When the solution search is failed, it is hard to tell that the solution technique is failed because of not enough search attempts or the non-existence of a solution.

Second, the non-uniqueness of inverse solutions requires some additional capabilities to a desirable solution technique: not only the capability of successfully identifying multiple solutions but also assessing the multiple solutions, e.g., which solution is more likely than others or which solution is physically impossible, etc.

Third, if an inverse solution does not depend on data continuously, the inverse solution becomes unstable. The continuous dependency of inverse solutions on data ensure that a small change in the data cause only a small change in the inverse solutions. On the contrary, if the dependency is discontinuous, a small change in the data can cause a large change in the inverse solutions. Figure 3 shows such a situation.

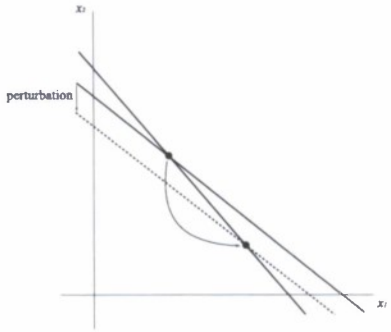


Figure 3: An Example of Unstable Inverse Solutions [40]

A system of two linear equations with two independent variables is shown in the two-dimensional inverse solution space. When a dependent variable contains a small error, which is manifested as a small perturbation of the corresponding line, the solution of the system of linear equations jumps one position to the other drastically, compared to the small magnitude of the perturbation. The sensitivity of a system of linear equations can be measured with the condition number of the matrix expressing the system of linear equations. Consider a system of linear equations in matrix form

$$\mathbf{y} = A\mathbf{x}$$

The condition number is the ratio of the largest singular value of the matrix A to the smallest one [14]. The matrix A can be viewed as the sensitivities between the variables (x) and the responses (y). In this case, the condition number can provide a measure of how influential the variables are relative to each other in describing the responses. This characteristic is especially important when dealing with data containing some errors such as measurement noise, which is always the case in the real world. As shown in Figure 3, a small error in a dependent variable caused by measurement noise can lead to large deviations in independent variables in the inverse problem.

3 Literature Review

Current approaches to the status matching problem as well as other potential solutions were identified through a literature search. The general subject of matching computer models to real data is covered thoroughly by Kennedy and O'Hagan [23]. While the subject of engine modeling itself, cycle decks in particular, has received little attention in the published literature, several valuable sources are available through the NATO AGARD and RTO publications. In addition, a rich source of reference material may be found in the related field of engine health monitoring and diagnostics. The subject of matching computer models to test data has also received some attention in the area of hydrological modeling. Finally, the field of numerical optimization was explored as a possible source for potential solutions to the status matching problem. These sources, along with a brief description of each method, are discussed in turn below.

3.1 AGARD Publications

One of the earliest descriptions of the status matching problem may be found in Habrard [15]. Habrard formulated the problem by superposing over the usual Newton-Raphson cycle balance an additional iteration comprising the (dependent) error terms for the measured data and (independent) tuning parameters or modifiers to adjust the component maps. Habrard made several recommendations that are characteristic of modern approaches:

- Additional estimated parameters should be included in the tuning equations in addition to the directly measured parameters
- The tuning equations should be augmented with weighting functions to account for the relative level of confidence attributed to the measured or estimated dependents
- He addressed the problem that the number of modifiers available may exceed the number of dependent parameters. The solution method he recommended was equivalent to the singular value decomposition (SVD) method described in later sections of this report.

3.2 Engine Diagnostics Literature

The engine diagnostics and health monitoring problem is closely related to the status matching problem. There is an extensive body of literature related to analytical enginc diagnostics; a good general review is available in Li [26].

The subject of analytical engine diagnostics begins with Urban's pioneering work, GPA (Gas Path Analysis) [38]. Changes in the health of an engine are manifested as changes in measurable engine parameters. GPA uses these sensitivities in matrix form, called the *influence coefficient* matrix written as follows:

$$H = \begin{pmatrix} \frac{\Delta z_1}{\Delta x_1} & \frac{\Delta z_1}{\Delta x_2} & \cdots & \frac{\Delta z_1}{\Delta x_n} \\ \frac{\Delta z_2}{\Delta x_1} & \frac{\Delta z_2}{\Delta x_2} & \cdots & \frac{\Delta z_2}{\Delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta z_m}{\Delta x_1} & \frac{\Delta z_m}{\Delta x_2} & \cdots & \frac{\Delta z_m}{\Delta x_n} \end{pmatrix}$$

where z_i is a measurable engine parameter and x_j is an engine performance modifier. The influence coefficient matrix is a linear approximation of the real, possibly nonlinear, relationships between the measurable engine parameters and the engine performance modifiers. When some measurable parameters are available, the engine performance modifiers can be calculated by the inversion of the influence coefficient matrix or the weighted least squares (WLS) method. Urbans's work inspired many other researchers, and there are various extensions of the original GPA. In late 1970s, GE Aircraft Engines developed a similar program called TEMPER [10]. Urban's GPA was followed by many other similar research works.

Another linear approach in engine status matching is the use of the Kalman filter [21]. The Kalman filter is a recursive filter that estimates the state of a linear dynamical system from a series of noisy measurements. The Kalman filter consists of two phases: predict and update. In the predict phase, the current state is predicted from the state at the previous time step. In the update phase, the measurement at the current time step is used to correct the predicted state. In the 1980s, Rolls Royce developed COMPASS (COndition Monitoring and Performance Analysis Software System) using the Kalman filtering technique to estimate turbofan engine health parameters and sensor bias from measurements [30]. To improve the performance of the Kalman filter in diagnostics of gas turbine engine, the Kalman filter is used in many different ways, for example, the constrained Kalman filtering [32] and a bank of Kalman filters [24]. The extended Kalman filter (EKF) [33] and unscented Kalman filter (UKF) [19] are nonlinear extensions of the Kalman filter.

3.3 Optimization Methods

The discipline of numerical optimization may also provide a source for solutions to the status matching problem. Historically, there have been efforts to estimate engine performance modifiers by converting the inverse problem to one of optimization. In these efforts a numerical optimizer finds engine performance modifiers that make the result of a thermodynamic engine model as close as possible to measured engine performance parameters. "Closeness" to measured performance is typically calculated using a single objective function. The objective function combines all available performance measurements into a single measure of fitness. It is then the job of the optimization routine to find the set of modifiers to maximize the overall fitness of the objective function [9].

Most optimization techniques can typically be grouped into two categories: 1) line search or gradient-based techniques and 2) heuristic or stochastic searches. The gradient-based methods perform extremely well for convex problems which have a unique global optimum, but may stop searching prematurely in the presence of many local optima. Stochastic methods, although usually requiring more computer resources, are able to avoid becoming stuck at a local optimum [39].

Because of the nonlinear relationships between measurable engine parameters and engine performance modifiers, the genetic algorithm (GA) is preferred over gradient-based optimization methods [43]. GA is also of interest for its potential to add intelligence to otherwise random search methods such as Filtered Monte Carlo. Rather than blindly sample a solution space, the algorithm drives toward a more optimum solution at each iteration by combining attributes of multiple "genetically fit" solutions. This property, called crossover, combined with mutation, can find settings of the performance modifiers that match the responses quite accurately. For more detailed explanation of genetic search techniques, the reader is referred to [39]. Although GA has better chance to find the global optimum than gradient-based methods, the global optimality of its solutions is still not guaranteed. Furthermore the genetic algorithm is computationally demanding, requiring a large initial population and many iterations.

3.4 Surrogate Modeling

In the above optimization approach a thermodynamic engine model is used to simulate the engine performance. Instead of using a thermodynamic engine model, there have been efforts to use lower fidelity surrogate models or AI (Artificial Intelligence) techniques to interpret the measurable engine parameters and relate them to the engine performance modifiers. An advantage to these methods is that complex computation/optimization can be performed on the surrogates which are typically polynomial expressions versus a much slower computer cycle model. This fact makes surrogate models ideal for Monte Carlo methods where it is not uncommon to perform many thousands of function calls. These techniques learn the mappings between the measurable engine performance parameters and the engine performance modifiers from available data. The trained mappings are then used to predict engine performance modifiers when new measurable engine performance parameters arrive. A common surrogate modeling technique and one that will be investigated in this research is response surface equations (RSEs). RSEs are perhaps the fastest type of surrogate to create through a simple least squares regression, but they suffer from an inability to represent nonlinear design spaces. While thermodynamic cycle behavior across the entire flight regime is highly nonlinear, RSEs may be used to represent a small region of the space such as a single setting of ambient condition and throttle position.

While the literature search yielded alternative techniques including those from the field of AI such as neural networks [44], fuzzy logic [12], and expert systems [37], these methods were not pursued as viable means of solving the status matching problem. Although more sophisticated surrogate modeling techniques like neural networks map complex nonlinear relationships quite well, they are difficult to train as they typically require their own optimization algorithm and significant computational resources. Therefore, the time required to merely generate the networks negates any possible advantage gained through their ability to model nonlinear space. What is probably worse, there is a severe lack of transparency when using neural networks so that it is difficult to gain insight into the behavior of multidimensional spaces. Fuzzy logic is often criticized with regards to scalability. Although fuzzy logic has been used in control applications for home appliances, there are few publications regarding the use of fuzzy logic in the real world [11]. Expert systems are mostly used for solving qualitative problems.

3.5 Other Statistical Methods

In addition to the AI techniques mentioned above, Bayesian networks have been paid great deal of attention recently. The Bayesian network technique is a framework combining graph theory and probability theory. A Bayesian network for engine diagnostics probabilistically models the relationships between measurable engine parameters and engine performance modifiers. When the Bayesian network is used in diagnostics, the measured data is entered into the network, and the engine performance modifiers are inferred. Breese et al. [7], Mast et al. [28], and Romessis and Mathioudakis [31] applied the Bayesian network technique for diagnostics of industrial or aircraft engine. Lee [25] proposed a use of multiple Bayesian networks to increase the accuracy and robustness of estimates of engine performance modifiers. Although it has shown a great potential for its usage, the Bayesian network technique is often criticized for its mathematical complexity and computational burden.

It is interesting that singular value decomposition (SVD) and regularization, which are the most widely known inverse problem solution techniques in the fields of medical imaging and statistics, are rarely used in gas turbine diagnostics.

Singular value decomposition (SVD), developed in linear algebra, is a technique for solving underdetermined system of linear equations. An underdetermined system of linear equation has an infinite number of solutions. Among these solutions the minimum 2-norm solution is always unique [6]. SVD finds the unique, minimum 2-norm solution. However, the minimum norm solution may not be physically meaningful in practical problems.

Regularization is a classical approach for stabilizing unstable inverse solutions. Regularization methods seek a stable, approximate solution by adding an extra constraint on inverse solutions. Two widely used constraints are the 1-norm or 2-norm of the inverse solution. Regularization using these two constraints is called the *Lasso* [35] and *Tikhonov regularization* [36], respectively.

3.6 Hydrological modeling literature

Among the many applications discussed by Kennedy and O'Hagan [23] is that of hydrological simulation, dating back to the pioneering work of Beven and Binley [4]. The Generalized Likelihood Uncertainty Estimation (GLUE) method developed by Beven and Binley is similar in many respects to the Filtered Monte Carlo (FMC) method described in the following sections.

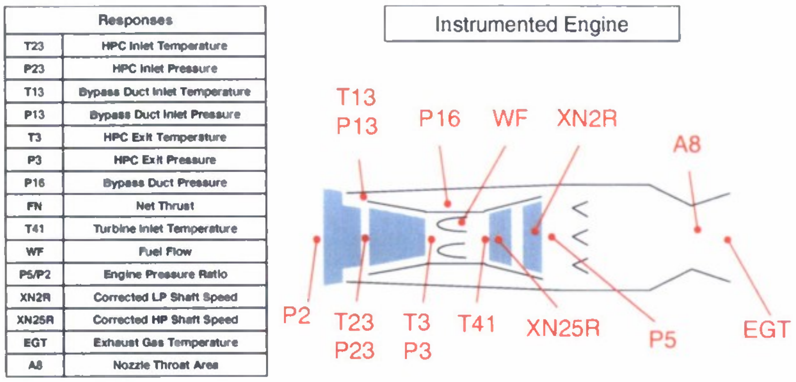
4 Algorithm Selection

Once the literature search identified a list of potential solutions to the status matching problem given the time and scope of the project, the next step was to develop a procedure for selecting, refining, and further developing one or more of those solutions into a practical, easy-to-use method. A subset of algorithms was identified from the literature for further investigation and direct application on a simple thermodynamic cycle model calibration problem. This way, each algorithm's performance could be compared side-by-side and the necessary refinement and/or rejection of the method could take place. The methods chosen for further research were the following: FMC, genetic algorithm, SVD, regularization, Kalman filters (EKF and UKF), and Bayesian networks. These methods showed the most potential for obtaining accurate estimations for the performance modifiers given the non-linear and otherwise complex behavior of the inverse problem. For those methods that require calculation of a sensitivity matrix (EKF, SVD, and Regularization), a local iterative technique of solving for the performance modifiers was adopted. In other words, the local derivative matrix (Jacobian) was calculated at an initial guess point. The algorithm was executed in order to update the initial guess to a better estimate. The Jacobian was then calculated for the new estimated point and the procedure was repeated until convergence. This method can be likened to trust region optimization and imposes stability on an inherently nonlinear problem. In addition to the particular algorithms, RSEs were investigated as a potential enabler of faster computation and transparency.

In order to qualitatively assess the performance of each solution technique, a set of metrics was adopted. These metrics were arrived at using engineering judgment as well as input from potential end-users of the product at Arnold Engineering Development Center (AEDC) in Tullahoma, TN. The first and perhaps most obvious metric was the ability for the method to match the given measurement data. The performance modifiers obtained from a given method ought to produce a resulting data point that matches as closely as possible the measured data. This can be stated quantitatively simply as a summation of percent errors between the measurement data and what an individual method predicts. The next metric used to evaluate the solution methods is the overall ease of implementation and use. It was an original goal of the project to develop a method that could be used not necessarily by a performance engineer intimately familiar with a given turbine engine or even statistician well-versed in complex inverse problem theory. The method must be sophisticated enough to satisfactorily solve the complex problem and yet simple enough to be used by anyone on any engine. A qualitative representation of this metric is the time needed to set up or code the algorithm as well as the number of additional pieces of information required (parameter constraints, covariance, weighting, etc.) to produce reliable results. The final metric used to compare methods was their respective computational demands. The time and effort required to create a status deck using current status matching techniques is a significant disadvantage. The new method must use minimal computational resources to match multiple data points quickly and accurately. This can be represented as the time to match a single point, the number of thermodynamic cycle cases/runs, or amount of function calls/iterations.

4.1 Sample Problem

Once the metrics were identified, a sample problem was set up to serve as a test bed for each potential status matching technique. The test bed chosen was a generic mixed flow turbofan (MFTF) engine developed using NASA's Numerical Propulsion System Simulation (NPSS). NPSS is an object-oriented modeling environment widely used throughout industry and the USAF. With NPSS, the engine is modeled as an assembly of component "elements". Each element contains "sockets" for the component maps and additional empirical factors. "Audit" modifiers are available for adjusting the component representations. The scripting language in NPSS allowed for easy implementation of each solution method. A drawing of the sample engine model is presented in Figure 4, annotated to show the simulated instrumentation locations. The figure also defines the variable names.



Responses	
T23	HPC Inlet Temperature
P23	HPC Inlet Pressure
T13	Bypass Duct Inlet Temperature
P13	Bypass Duct Inlet Pressure
T3	HPC Exit Temperature
P3	HPC Exit Pressure
P16	Bypass Duct Pressure
FN	Net Thrust
T41	Turbine Inlet Temperature
WF	Fuel Flow
P5/P2	Engine Pressure Ratio
XN2R	Corrected LP Shaft Speed
XN25R	Corrected HP Shaft Speed
EGT	Exhaust Gas Temperature
A8	Nozzle Throat Area

Figure 4: Engine Model and Nomenclature for Sample Engine Model

Since the cycle model did not represent an actual engine, no "noisy" data was readily available. In order to simulate noisy engine measurement data, a single point of the baseline MFTF model was run with a select set of notional performance modifiers applied. This produced a set of responses to match using each technique beginning with the baseline cycle model. The responses themselves were chosen based on what typically is available from a moderately instrumented turbine engine. These are given along with their simulated measurement values in Table 1.

Table 1: Simulated Measurements for MFTF

Measurement	Measurement Description	Value	Units
T23t	HPC Inlet Total Temp	663.688	°R
P23t	HPC Inlet Total Pressure	30.457	psi
T13t	Bypass Duct Inlct Total Temp	687.11	°R
P13t	Bypass Duct Inlet Total Pressure	34.409	psi
T3t	HPC Exit Total Temp	1390.516	°R
P3t	HPC Exit Total Pressure	320.765	psi
P16	Bypass Duct Static Pressure	32.332	psi
FN	Net Thrust	20726.7	lb_f
T41	Turbine Inlet Temp	3261.9	°R
WF36	Fuel Flow	3.518	lb/s
EPR	Enginc Pressure Ratio	2.378	--
PCN2R	Corrected LP Shaft Speed	99.9998	%
PCN25R	Corrected HP Shaft Speed	98.6145	%
EGT	Exhaust Gas Temp	1197.23	°R
A8	Nozzlc Throat Area	716.9	in^2

The performance modifiers used to generate these measurements were chosen at random within a range selected to be representative of the error in a baseline cycle model. Therefore, a modifier value of one indicates the baseline cycle. All values were chosen to be slightly less than one to simulate a degraded engine. These are given in Table 2.

Table 2: Simulated Performance Modifiers for MFTF

Modifier Name	Modifier Description	Value
Fan.eff	Fan Efficiency Scalar	0.986
Fan.Wc	Fan Corrected Flow Scalar	0.996
HPC.eff	HPC Efficiency Scalar	0.976
HPC.Wc	HPC Corrected Flow Scalar	0.986
Burner.dPqP	Combustor Pressure Drop Scalar	0.996
LPT.eff	LPT Efficiency Scalar	0.986
LPT.Wp	LPT Corrected Flow Scalar	0.976
HPT.eff	HPT Efficiency Scalar	0.976
HPT.Wp	HPT Corrected Flow Scalar	0.976
Nozzle.Cfg	Thrust Coefficient Scalar	0.981
Duct1.dPqP	HPC Inlet Duct Pressure Drop Scalar	0.996
Duct2.dPqP	HPC Exit Duct Pressure Drop Scalar	0.976
Duct3.dPqP	LPT Exit Duct Pressure Drop Scalar	0.991
Duct4.dPqP	Nozzle Inlet Pressure Drop Scalar	0.981
Duct5.dPqP	Bypass Duct Pressure Drop Scalar	0.986

An advantage to using this approach, while contrived, is that the values of the performance modifiers were known in advance. This provided an additional metric with which to rank the solution methods: ability to reproduce the initial set of modifiers given only the noisy set of responses. This would be infeasible with real engine test data or actual baseline cycle model because the correct values of the performance modifiers would not be known beforehand.

4.2 Algorithm Selection Results and Conclusions

The first results and conclusions to be drawn were those concerning the use of RSEs. The goal of the investigation was to determine if their potential to replace the more complex cycle model was realizable. It was determined that it was not. The time savings gained by executing the polynomial expressions was lost in the generation of the RSEs themselves, which remains a very manual process. This can be seen in Figure 5 in the comparison between the time it takes to run an equal number of cases on the NPSS cycle model versus generating and executing RSEs.

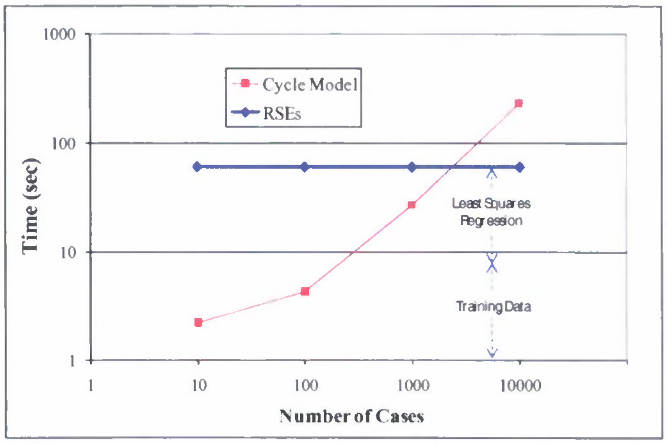


Figure 5: Cycle Model vs. RSE Execution Time

While the hypothesis that linearization of the cycle model about a single data point (single set of inlet conditions and throttle setting) was appropriate, this fact still required a set of RSEs for every point in the flight map. Furthermore, to ensure optimality, the final solution (or set of solutions in the case of FMC or Bayesian networks) still required validation within the cycle model itself. From this point on, the plan for each algorithm moving forward was for it to be implemented using the cycle model rather than a surrogate.

The next step was to test and evaluate each method according to the chosen metrics. A discussion of each metric will follow along with some noteworthy observations and general comments in order to provide the justification for down-selection to two final methods. First, closeness to measurement data is shown in Table 3.

Table 3: Converged Results and Closeness to Measurement Data: EKF and UKF

Response	Target	EKF Results	% Diff	UKF Results	% Diff
T23	663.688	663.69	0.001%	663.69	0.000%
P23	30.457	30.46	-0.003%	30.46	0.001%
T13	687.11	687.11	0.000%	687.11	0.000%
P13	34.409	34.41	-0.003%	34.41	0.003%
T3	1390.516	1390.50	-0.001%	1390.53	0.001%
P3	320.765	320.75	-0.005%	320.77	0.003%
P16	32.332	32.34	0.012%	32.33	0.004%
Thrust	20726.7	20726.80	0.000%	20726.63	0.000%
T41	3261.9	3262.20	0.009%	3261.72	-0.006%
WF36	3.518	3.52	0.028%	3.52	0.005%
EPR	2.378	2.38	0.000%	2.38	-0.007%
XN2	99.9998	100.00	0.000%	100.00	0.000%
XN25	98.6145	98.61	-0.005%	98.61	0.000%
EGT	1197.23	1197.26	0.003%	1197.25	0.001%
A8	716.9	716.93	0.004%	716.90	0.000%

While the preceding table shows only EKF and UKF, the results illustrate how all the methods could be evaluated. This result was surprisingly consistent for all methods: they could match the measurement data with a high degree of accuracy. This was especially true for those that required calculation of the coefficient matrix/Jacobian: EKF, Regularization, and SVD. Using information about the model, in this case the sensitivities of the measurements to changes in the performance modifiers, proved to yield very accurate final solutions with respect to the outputs. The accuracy in predicting the performance modifiers proved to be the opposite case, as seen in following table.

Table 4: Converged Performance Modifier Values for EKF and UKF

Modifier	Target	EKF	% Diff	UKF Results	% Diff
Fan.eff	0.986	0.98602	0.002%	0.98600344	0.000%
Fan.Wc	0.996	0.995908	-0.009%	0.9959854	-0.001%
HPC.eff	0.976	0.976014	0.001%	0.97599974	0.000%
HPC.Wc	0.986	0.986077	0.008%	0.98602274	0.002%
Burn	0.996	0.997904	0.191%	0.96015892	-3.599%
LPT.eff	0.986	0.999041	1.323%	1.00260338	1.684%
LPT.Wp	0.976	0.990826	1.519%	0.99500236	1.947%
HPT.eff	0.976	0.968183	-0.801%	0.96516456	-1.110%
HPT.Wp	0.976	0.976176	0.018%	0.97465093	-0.138%
Cfg	0.981	0.981106	0.011%	0.98100936	0.001%
Duct1	0.996	1.001182	0.520%	0.99396482	-0.204%
Duct2	0.976	1.001486	2.611%	1.02337535	4.854%
Duct3	0.991	0.995698	0.474%	0.98508077	-0.597%
Duct4	0.981	0.995629	1.491%	1.01233622	3.194%
Duct5	0.986	0.994732	0.886%	0.98674393	0.075%

The difference in converged values is not insignificant even though both points provide almost identical measurements. This highlights a non-trivial difficulty in solving inverse problems and one that appeared in every tested method: there may be infinitely many solutions that satisfy convergence criteria within a given tolerance. This will be addressed to a degree in following sections when refining the selected methods.

In terms of computation time, the EKF had a slight advantage over the UKF. While the EKF still required calculation of the local derivatives, UKF required many more iterations of the filter before convergence. In general however, the computation times for all methods tested with the exception of the stochastic methods (FMC and GA) were on the same order of magnitude. The computation time for FMC and GA depended directly on the number of performance modifiers used and desired resolution of the random search. GA especially took a significant amount of time longer than FMC yet provided comparable results. The implementation of the GA also introduced a level of complexity in coding and execution that provided no incentive for choosing it over the simpler FMC method.

The final metric used to assess the performance of the status matching techniques was overall simplicity in implementation and use. While the other two criteria were valuable in gaining insight into the methods themselves, this was the only one that could identify methods with distinct advantages over others. Both types of Kalman filter, for example, require many pieces of information and assumptions to utilize to their full potential. This *a priori* knowledge is not critical to obtaining a solution using a Kalman filter. However, much of the information that normally makes the Kalman filter a powerful tool for state prediction is absent on a real cycle model calibration problem: covariance between performance modifiers, distribution of measurement error, etc. This "bare-bones" approach essentially reduces the Kalman filter to a weighted least squares solution. This need for reduction of each method to a simpler form by imposing minimal prior knowledge led to a rejection a many potential solutions on the grounds that this knowledge may not be available on a real status matching problem.

The advantages and disadvantages of each algorithm are summarized in Table 5 below.

Table 5: Algorithm Advantages and Disadvantages

	Advantages	Disadvantages
Filtered Monte Carlo Simulation	1. Able to find global optimum 2. Intuitive/easy to implement procedure 3. Multiple solution vectors	1. Difficult to obtain parameters that have small effect on responses 2. High computation time 3. Solution highly dependent on initial distribution assumption 4. May need multiple iterations to obtain converged solution 5. High dimensional problem needs many cases to adequately define space
Bayesian Network	1. Probabilistic 2. Adaptive or dynamic correction possible 3. Allows analysis with missing data 4. Based on posterior probabilities leading to better predictive results	1. Difficult to obtain parameters that have small effect on responses 2. Generation of surrogates a time-consuming and tedious process 3. Additional uncertainty introduced with surrogate models 4. May need to build at network at each flight condition 5. Difficult to assign prior distributions and CPDs 6. Run time exponentially increases with number of independent variables
Kalman Filter	1. Time to convergence (EKF) and few iterations 2. Well behaved convergence if appropriate assumptions 3. Similar in applicability to Bayesian Methods 4. Easy to implement and its ability well proven in several industrial applications 5. UKF avoids Jacobian calculation, but at an additional computational cost.	1. Difficult to obtain parameters that have small effect on responses 2. Solution dependent on initial assumptions of covariance and noise 3. Covariance and noise dependency not intuitive 4. Error associated with Jacobian calculation using finite difference 5. Single solution vector
Genetic Algorithm	1. Able to find global optimum 2. Multiple solution vectors 3. Avoids tendency to over-predict results as with MCS	1. Difficult to obtain parameters that have small effect on responses 2. High number of NPSS runs leads to long computation time 3. Solution depends on resolution of discretization

After weighing each metric and eliminating those that had unacceptable characteristics or offered no advantages, two methods were chosen: filtered Monte Carlo and Singular Value Decomposition. While each may be used on its own as a standalone solution to status matching, they can also complement each other; FMC is the simplest way to examine the entire solution space, while SVD is a reliable method for an inexperienced user to find a unique solution. Each method is described in detail in the following sections.

5 Description of Selected Algorithms

The Filtered Monte Carlo and Singular Value Decomposition methods are described in the following sections.

5.1 Filtered Monte Carlo (FMC) Method

The Filtered Monte Carlo method is carried out by simply assigning random values to each of the modifiers, running the model with those values, and comparing the model outputs to the corresponding test measured parameters. The procedure is illustrated graphically in Figure 6.

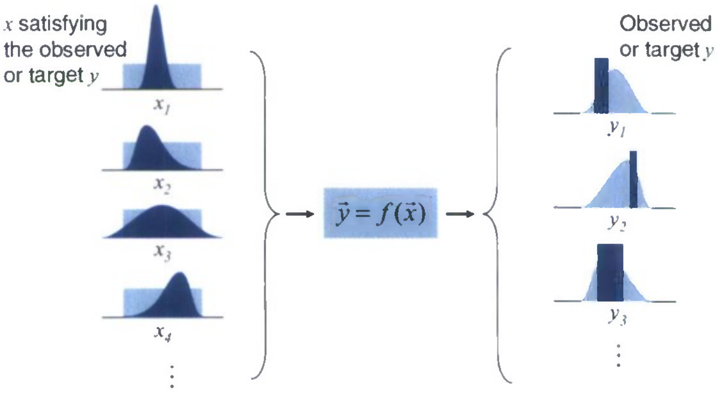


Figure 6: Schematic of Filtered Monte Carlo process

The input modifiers may be assigned values over any specified range and using any specified probability distribution function desired by the analyst. Input sets that produce outputs matching the test data to within some specified tolerance are retained, and the other input sets are discarded. The process may be repeated any number of times to narrow down to the most probably solution.

There are two ways to interpret and apply the Filtered Monte Carlo results. First is the "probabilistic" method, whereby the analyst filters the results to a tolerance sufficient to obtain reasonable probability distributions, and then selects the mean values as "the" solution. A refinement to this approach might be to examine the joint probability space to determine a "most probable point" solution. The second method is the "deterministic" method, whereby the analyst filters the data very tightly to eliminate all but a few solutions. The analyst then examines each solution in detail to select the "best" one. However, it should be noted that the solution lying closest to the measured value may not be the most probable solution. For this reason the probabilistic interpretation is usually preferred over the deterministic interpretation.

5.2 Singular Value Decomposition (SVD)

SVD was first introduced to meteorology in a 1956 paper by Edward Lorenz [Lorenz], in which he referred to the process as empirical orthogonal function analysis. Today, it is also commonly known as principal component analysis (PCA). All three names are still used, and refer to the same set of processes. SVD methods deal with solving difficult linear-least squares problems. SVD method when used for status matching implements a linearization process.

They are based on the following theorem of Linear Algebra:

"Any M x N matrix A whose numbers of rows M is greater than or equal to its number of columns N can be written as the product of an M x N column orthogonal matrix U , an N x N diagonal matrix W of singular values and the transpose of an N x N orthogonal matrix V:

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} = \begin{pmatrix} & & \\ & U & \\ & & \end{pmatrix} \begin{pmatrix} w_1 & & \\ & \ddots & \\ & & w_n \end{pmatrix} \begin{pmatrix} & & \\ & V & \\ & & \end{pmatrix}$$

Qualitatively the U matrix represents a vector basis for the most relevant information in the system while the eigenvalues w_i represent the variability in the information.

SVD plays a very important role in linear algebra. It has applications in solving least squares problems, in computing the pseudoinverse, in computing the Jordan canonical form, in solving integral equations, in digital image processing, and in optimization. Many of the applications often involve large matrices. It is therefore important that the computational procedures for obtaining the SVD be as efficient as possible. The basis of the most popular modern singular value decomposition algorithms consists of two phases. In the first phase one constructs two finite sequences of transformations

$$P^k, k = 1,2,....n$$

and

$$Q^k, k = 1,2,....n-2,$$

such that $$P^{n}...P^{1}AQ^{1}...Q^{n-2} = \left[\begin{array}{cccc} X & X & & \\ & \bullet\bullet & & \\ & & \bullet\bullet & \\ & & & \bullet\bullet \\ \hdashline & & O & \\ & & & \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c} X \\ \bullet \\ \bullet \\ \bullet \end{array}}\right\} n \\ \vphantom{O} \\ \vphantom{O} \end{array}$$ $= J^{0}$, an upper bidiagonal matrix.

Specifically, P^{i} zeros out the subdiagonal elements in column i and Q^{j} zeros out the appropriate elements in row j.

Because all the transformations introduced are orthogonal, the singular values of J^{0} are the same as those of A. Thus, if

$$J^{0} = G\Sigma H^{T}$$

is the SVD of J^{0}, then

$$A = PG\Sigma H^{T}Q^{T},$$

so that

$$U = PG, V = QH.$$

The second phase is to iteratively diagonalize J^{0} by the QR method so that $J^{0} \rightarrow J^{1} \rightarrow ... \rightarrow \Sigma$, where $J^{i+1} = (S^{i})^{T} J^{i} T^{i}$ where S^{i} and T^{i} are orthogonal.

The matrices T^{i} are chosen so that the sequence $M^{i} = (J^{i})^{T} J^{i}$ converges to a diagonal matrix, while the matrices S^{i} are chosen so that all J^{i} are of bidiagonal form. The products of the T^{i} and the S^{i} are exactly the matrices H^{T} and G^{T}, respectively

The computation is usually implemented in a computer program as follows: Assume for simplicity that the matrix A can be destroyed and that U can be returned in the storage for A. In the first phase the P^{i} are stored in the lower part of A, and the Q^{j} are stored in the upper triangular part of A. After the bidiagonalization, the Q^{j} are accumulated in the storage provided for V, the two diagonals of J^{i} are copied to two other linear arrays, and the P^{i} are accumulated in A. In the second phase, for each i, S^{i} is applied to P from the right, and $(T^{i})^{T}$ w is applied to Q^{T} from the left in order to accumulate the transformations.

SVD can not fail to give an answer in theory; it can give results for over-determined, under-determined, or singular matrices. For under-determined problems, SVD finds the Bayesian solution which minimizes error residuals while simultaneously being closest to the prior guess for the independent variables. For over-determined problems, SVD finds the least squares solution that minimizes SSE over all residuals.

SVD is a method to solve systems of linear equations. In order to use svd for status matching, which is nonlinear, the linear equations are successively linearized. This procedure is illustrated in Figure 7. The SVD method is an "outer loop" surrounding the "inner loop" which is the standard Newton-Raphson iteration to balance the cycle (i.e., to satisfy the continuity and work requirements) for the current set of modifiers computed by the "outer loop".

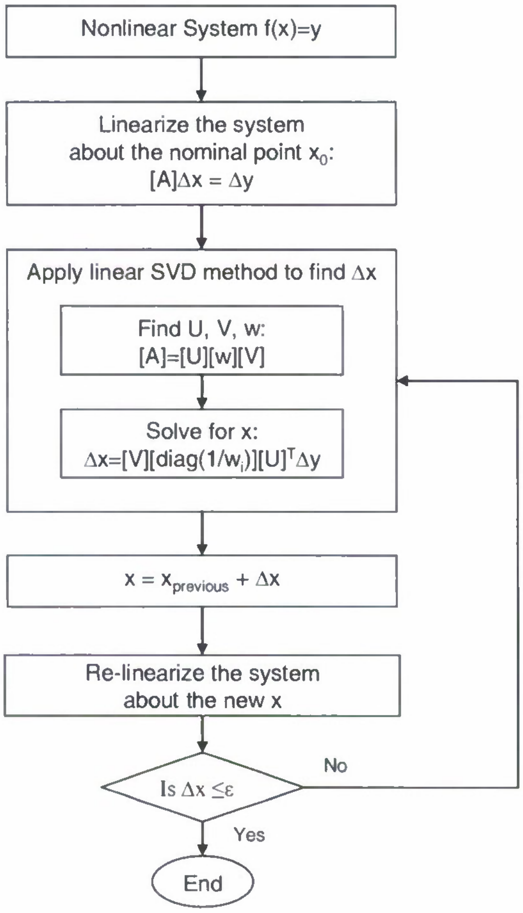

Nonlinear System f(x)=y
Linearize the system about the nominal point x_0: $[A]\Delta x = \Delta y$
Apply linear SVD method to find Δx
Find U, V, w: $[A]=[U][w][V]$
Solve for x: $\Delta x=[V][\mathrm{diag}(1/w_i)][U]^T\Delta y$
$x = x_{previous} + \Delta x$
Re-linearize the system about the new x
Is $\Delta x \leq \varepsilon$
No
Yes
End

Figure 7: Flowchart of SVD Calculations

One refinement to the process is to make use of the eigenvalue information (w_i) to identify and remove unimportant modifiers from the SVD solution. If the eigenvalues are smaller than a predetermined limit, they are zeroed out so that the values of the corresponding modifiers are no longer updated. This procedure helps avoid Type I errors of including unnecessary modifiers.

5.3 Measurement uncertainty considerations

From the perspective of status matching, uncertainty in the measured parameters, while important, is not necessarily a primary concern. It is usually much more important to determine whether a measurement is "bad" or invalid, either due to a malfunction of the sensor or due to poor placement of the sensor, making the measurement overly sensitive to the three-dimensional unsteady flow field within the engine. The identification of an invalid measurement must be made within the context of the other measurements which are accepted by the analyst to be valid. That is, one parameter may be observed not to "close" with the others while satisfying the conservation laws with the engine model. However in making such identifications, the analyst must keep in mind that the apparently invalid measurement may in fact be correct; the remaining parameters may exhibit only subtle shifts from their "normal" values within their measurement uncertainties. The experience and judgment of the analyst, perhaps combined with physical inspection of the hardware, may be required to resolve this type of problem. As such, it is considered beyond the scope of the current research, although it may be a fruitful area for future work.

In addition to sensor error, there are other sources of noise that affect the solution, such as the measured ambient conditions (e.g., T_{amb} and P_{amb} or altitude and Mach number). These sources must be included as part of the total uncertainty of a measurement. For example the total uncertainty of the thrust measurement F_n, including the effects of T_{amb} and P_{amb}, may be expressed as shown in the equation below:

$$\sigma^2_{fn\_equiv} = \sigma^2_{meas,fn} + (\sigma_{meas,Pamb} \frac{\partial F_n}{\partial P_{amb}})^2 + (\sigma_{meas,Tamb} \frac{\partial F_n}{\partial T_{amb}})^2$$

where σ is the standard deviation. The total measurement uncertainty may then be included in the selected analysis methods in a straightforward manner. In the case of the FMC method, for example, measurement uncertainties can be taken into account when selecting the ranges of values for filtering. In the case of the SVD method or any method that essentially solves a system of the form y=Ax, the measurement uncertainty can be incorporated by scaling or dividing through each row of the A and x matrices by the standard deviation assigned to the corresponding measurement.

6 Demonstration

As a proof of concept, real engine data from testing of Pratt and Whitney PW2037 turbofan engines was used to develop a status engine model from a baseline engine model using both the FMC and SVD methods. The results of the status matching provided by each method were then compared.

6.1 Description of the Data

Engine test data was provided by Delta Air Lines for use in this demonstration. The data consisted of 46 engines each tested at six different power settings called "bands". The six different bands made up a sea-level static power hook beginning with Band A at high power through Band F at idle. The engine data was collected during post-maintenance testing. It was not known how many operating hours were on the engines or what type of maintenance was performed.

6.2 Baseline Engine Model

The first step in analyzing the data was to develop a baseline engine model, which would then be tuned to the test data through use of the status matching processes. In the absence of a true baseline PW2037 model, one was developed specifically for this research. As was the case for the sample engine developed previously, the baseline model for the PW2037 was made using NPSS.

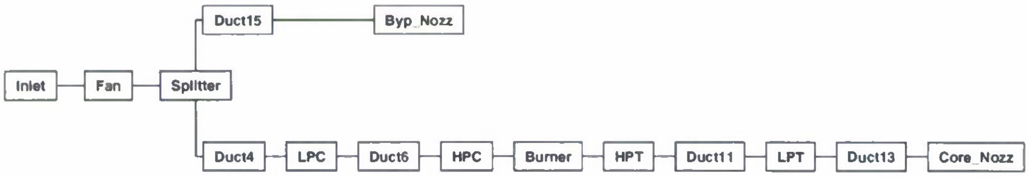


Figure 8: Schematic Diagram of NPSS Engine Model Components

The components of the NPSS engine model are shown schematically in Figure 8. Each turbomachinery component (Fan, LPC, HPC, HPT, and LPT) are represented by component maps and have audit modifiers for efficiency and corrected flow. The burner, the splitter, and each of the ducts have audit modifiers on their pressure loss characteristics. Finally, the two exhaust nozzles have audit modifiers on their discharge (or flow) coefficients.

The baseline model of the PW2037 engine was developed from data available in the public domain from the International Civil Aviation Organization's (ICAO) emissions databank [Aircraft]. Generic component maps and characteristics were scaled to match the ICAO data, resulting in a baseline model matching as closely as possible the engine certification data. Figure 9 shows the baseline model plotted with the ICAO data as well as the set of engine test data for the 46 PW2037s. The baseline model at similar inlet conditions to the test data is also shown on the figure.

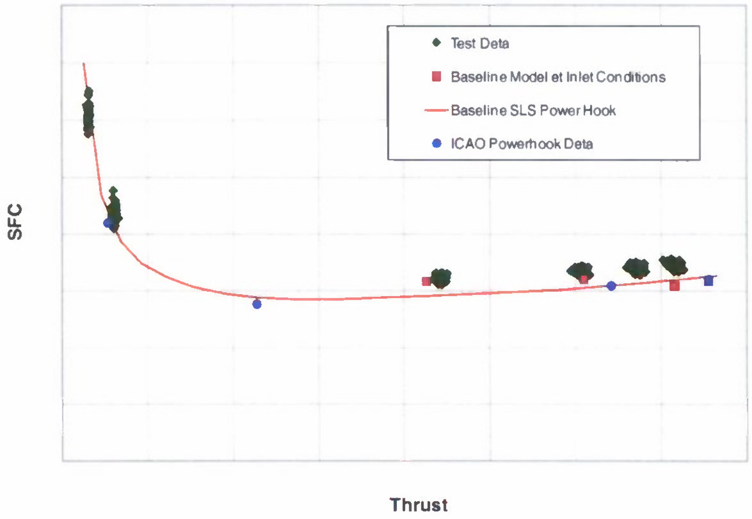


Figure 9: PW2037 Power Hook

As seen in the figure, the high power conditions show the most discrepancy between baseline and target. While data was available for six total throttle positions, only four were used in the demonstration. This was mainly a consequence of the baseline performance maps lacking the ability to predict behavior adequately for the lowest power settings. From here on, the demonstration will precede only with Bands A through D.

An additional test of matching ability could be seen in plotting various measurements against engine throttle setting for the four bands of interest. High pressure rotor speed, thrust, and fuel flow are shown below in Figure 10 as an example. In all cases, there is a significant discrepancy between the baseline model prediction and actual test data.

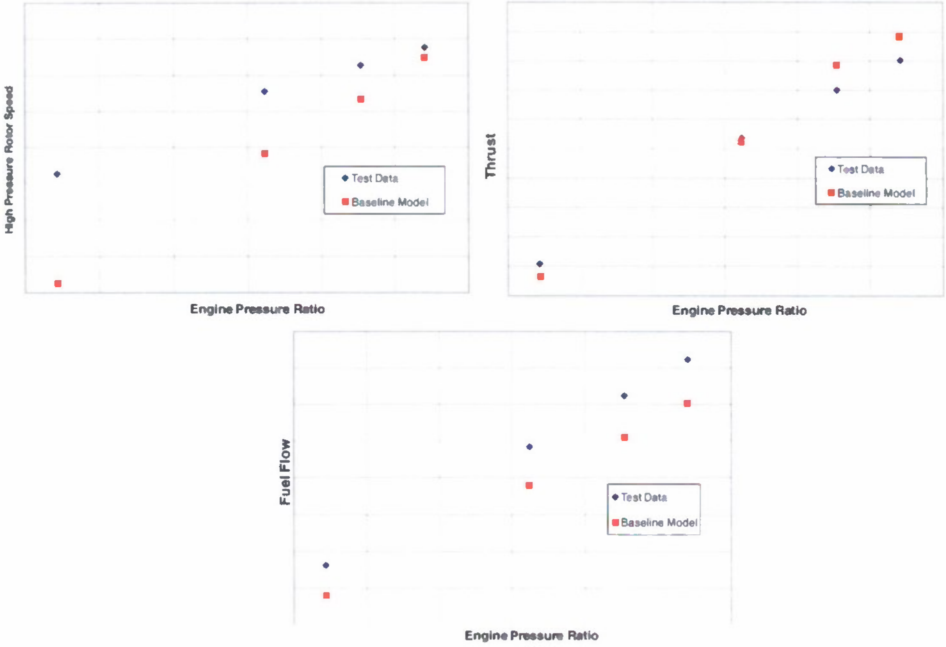
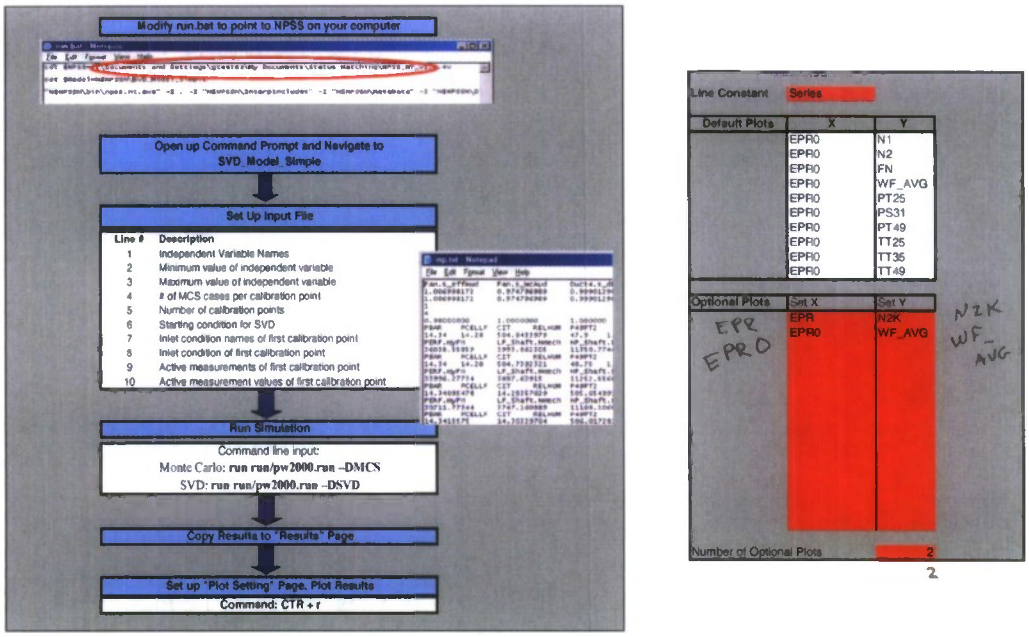


Figure 10: Baseline Model Comparison with Engine Test Data

In general, higher fidelity baseline models are typically used as the starting point for the status matching problem. The use of a generic model with estimated component performance maps certainly does not degrade the solution. In fact, the successful matching of cycle model predictions to test data on an initially poor representative model is a testament to the robustness of the status matching method.

6.3 User Interface

A user interface developed in Microsoft Excel Visual Basic for Applications (VBA) to address the need for a more automated status matching process. The "front page" of the interface, shown on the left in Figure 11, prompts the user through the steps necessary to set up and run the engine model. Setting up the input involves defining the independent and dependent variables, and mapping them from the test data sheet to the engine model input and output files. After the model is run, the results (computed modifiers) may be loaded into the results page. A plotting utility, shown on the right in Figure 11, automatically generates predefined plots of the results. The plotting utility allows the results to be compared to the original engine model as well as the test measured data.



Modify run.bat to point to NPSS on your computer

Open up Command Prompt and Navigate to SVD_Model_Simple

Set Up Input File

Line #	Description
1	Independent Variable Names
2	Minimum value of independent variable
3	Maximum value of independent variable
4	# of MCS cases per calibration point
5	Number of calibration points
6	Starting condition for SVD
7	Inlet condition names of first calibration point
8	Inlet condition of first calibration point
9	Active measurements of first calibration point
10	Active measurement values of first calibration point

Run Simulation

Command line input:
Monte Carlo: **run run/pw2000.run –DMCS**
SVD: **run run/pw2000.run –DSVD**

Copy Results to "Results" Page

Set up "Plot Setting" Page, Plot Results
Command: CTR + r

Line Constant: Series

Default Plots	X	Y
	EPR0	N1
	EPR0	N2
	EPR0	FN
	EPR0	WF_AVG
	EPR0	PT25
	EPR0	PS31
	EPR0	PT49
	EPR0	TT25
	EPR0	TT35
	EPR0	TT49

Optional Plots	Set X	Set Y
EPR	EPR	N2K
EPRO	EPR0	WF_AVG

N2K
WF_AVG

Number of Optional Plots: 2

2

Figure 11: User Interface

The user interface interacts with the engine model by several "wrapper" files which read in the data from a data file, set up the model according to the selected solution method, execute the model, and collect the results. By putting all the steps together under a single user interface, the data manipulation is greatly simplified and streamlined. This is a major contributor to speeding up the status matching process for an inexperienced user.

6.3.1 Interfacing Measurements

Instrumentation on each engine provided the measurements used to develop the status model. The available measurements used for status matching as well as those used to set the appropriate flight conditions for baseline model are listed in Table 6. It was assumed that throttle position controlled engine pressure ratio (LPT exit pressure divided by fan inlet pressure). Therefore, EPR was treated as an input to set the flight condition.

Table 6: PW2037 Measurements

Flight Condition	Measurement
Test Cell Inlet Temperature	Thrust
Outside Air Temperature	HP Rotor Speed
Relative Humidity	LP Rotor Speed
Fan Inlet Pressure	Fuel Flow
Engine Pressure Ratio (EPR)	HPC Inlet Pressure
	HPC Exit Pressure
	HPC Inlet Temperature
	HPC Exit Temperature
	LPT Inlet Temperature
	Exhaust Gas Temperature

An important aspect that should not be overlooked is how the physical measurements from instrumentation on the actual engine are mapped to variables in the cycle model on the computer. For this demonstration, it was assumed that there was a direct mapping. For example, the measurement for LPT inlet temperature corresponded exactly with the thermodynamic flow station immediately before the low pressure turbine. This assumption was used for every measurement and was based on the fidelity level of the baseline. Typically with a more sophisticated baseline, the probes and sensors used to obtain the measurements can be modeled and the appropriate losses and other effects can be recorded.

6.3.2 Interfacing independent variables

The next piece needed to complete the setup portion of the demonstration problem is identification of the performance modifiers used to match the test data. Without any prior knowledge of engine behavior, the complete set of available scalars was chosen. There were scalars for efficiency and flow on each rotating component (fan, LP/HP compressors, and LP/HP turbines). Each duct had an associated pressure loss expressed in terms of a scalar on baseline pressure loss. Lastly, there were pressure losses across the splitter, combustor, core, and bypass nozzles creating a total of 15 performance modifiers. The modifiers are summarized in Table 7.

Table 7: Component Performance Modifiers

Modifier Name	Description
Fan.s_effAud	Fan Efficiency Scalar
Fan.s_WcAud	Fan Flow Scalar
Duct4.s_dPqPaud	LPC Inlet Duct Pressure Drop Scalar
LPC.s_effAud	Low Pressure Compressor Efficiency Scalar
LPC.s_WcAud	Low Pressure Compressor Flow Scalar
Duct6.s_dPqPaud	HPC Inlet Duct Pressure Drop Scalar
HPC.s_effAud	High Pressure Compressor Efficiency Scalar
HPC.s_WcAud	High Pressure Compressor Flow Scalar
Burner.s_dPqPaud	Combustor Pressure Drop Scalar
HPT.s_WpAud	High Pressure Turbine Efficiency Scalar
HPT.s_effAud	High Pressure Turbine Flow Scalar
Duct11.s_dPqPaud	LPT Inlet Duct Pressure Drop Scalar
LPT.s_WpAud	Low Pressure Turbine Efficiency Scalar
LPT.s_effAud	Low Pressure Turbine Flow Scalar
Duct13.s_dPqPaud	LPT Exit Duct Pressure Drop Scalar
Duct15.s_dPqPaud	Bypass Duct Pressure Drop Scalar
Splitter.dPqP1	Splitter Bypass Stream Pressure Drop
Splitter.dPqP2	Splitter Core Stream Pressure Drop
Core_Nozz.s_CdTh	Core Discharge Coefficient
Byp_Nozz.s_CdTh	Bypass Discharge Coefficient

6.4 Results of SVD Method

The SVD method was used to compute modifiers for each of the test data points. To demonstrate the full process, i.e. including regression of the modifiers and incorporating them back into the model, a single representative engine was selected at random.

6.4.1 Representative Engine (SN# 716310)

A representative engine (SN# 716310) was selected from the sample of engine test data provided by Delta Air Lines. The match accuracy can be seen for the Band A data (100% power) in Figure 12. Note that the maximum error is around 0:025 % (not to be confused with 2.5 %). The resulting modifiers for all four bands are tabulated in Table 8.

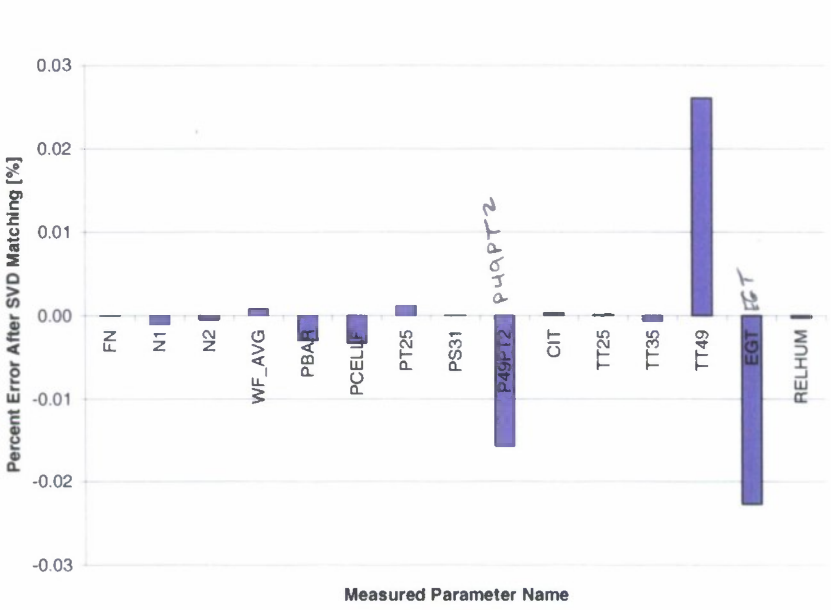


Figure 12: Percent Error for Engine SN# 716310 Band A (100% Power)

Table 8: SVD Results for Engine SN# 716310

Modifier Name	Band A	Band B	Band C	Band D
Fans_effAud.	1.01824	1.01926	1.02853	1.04701
Fans_WcAud.	0.96436	0.95476	0.95619	0.96058
Duct4s_dPqPaud.	1.00019	1.0001	1.00038	1.00049
LPCs_effAud.	1.09247	1.09066	1.08546	1.10765
LPCs_WcAud.	0.92826	0.93029	0.92174	0.91319
Duct6s_dPqPaud.	1.00043	1.00053	1.00058	1.00085
HPCs_effAud	1.01781	1.01943	1.01916	1.03046
HPCs_WcAud	0.9585	0.95128	0.93613	0.89866
Burners_dPqPaud	1.00207	1.00373	1.00298	1.00337
HPTs_WpAud	0.94299	0.94355	0.94451	0.94012
HPTs_effAud	0.93023	0.93063	0.92737	0.91543
Duct11s_dPqPaud	1.00021	1.00051	1.00035	1.00028
LPTs_WpAud	0.97852	0.97825	0.98343	0.99695
LPTs_effAud	0.94648	0.94317	0.93813	0.91292
Duct13s_dPqPaud	0.99612	0.99513	0.99519	0.99304
Duct15s_dPqPaud	1.0006	1.00065	1.00064	1.00042
SplitterdPqP1	0.01435	0.02625	0.03206	0.04691
SplitterdPqP2	0.05046	0.05318	0.05511	0.03482
Core_Nozzs_CdTh	1.00003	1.00002	1	1.00001
Byp_Nozzs_CdTh	1.00003	1.00002	1	1.00001

Due to the limited amount of data, the regression of the modifiers was carried out for demonstration purposes only and should not be considered as representative of the regression quality that is possible when more data is available. To demonstrate the process, each modifier was either fitted to a polynomial function of corrected fan speed (N1K) to account for the variation from band to band, or set to a constant if the variation across power bands was sufficiently small, as indicated in Table 9 below.

Table 9: Final Values for Status Deck – Engine SN# 716310

Modifier Name	Status Deck Value
Fans_effAud	Curve (function of N1K)
Fans_WcAud	0.9590
Duct4s_dPqPaud	1.0003
LPCs_effAud	Curve (function of N1K)
LPCs_WcAud	Curve (function of N1K)
Duct6_dPqPaud	1.0006
HPCs_effAud	Curve (function of N1K)
HPCs_WcAud	Curve (function of N1K)
Burners_dPqPaud	1.003
HPTs_WpAud	0.9428
HPTs_effAud	Curve (function of N1K)
Duct11s_dPqPaud	1.0003
LPTs_WpAud	Curve (function of N1K)
LPTs_effAud	Curve (function of N1K)
Duct13s_dPqPaud	0.9949
Duct15a_dPqPaud	1.0006
SplitterdPqP1	Curve (function of N1K)
SplitterdPqP2	Curve (function of N1K)
Core_Nozzs_CdTh	1.0000
Byp_Nozzs_CdTh	1.0000

The representative curve fits are shown in the Figures below.

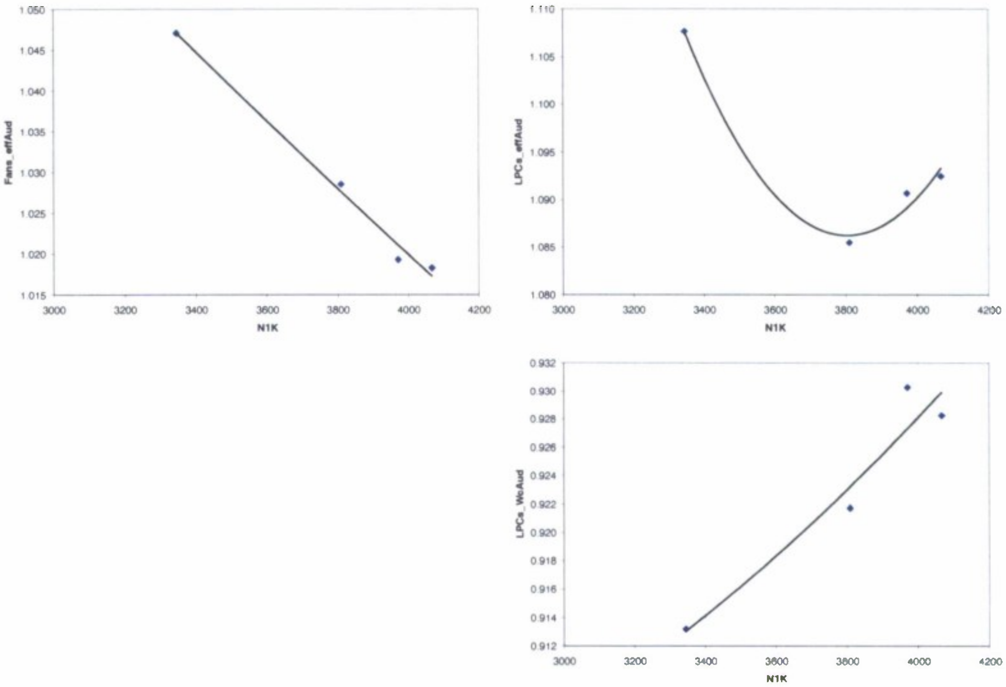


Figure 13: Modifier curve fits for Fan and LPC

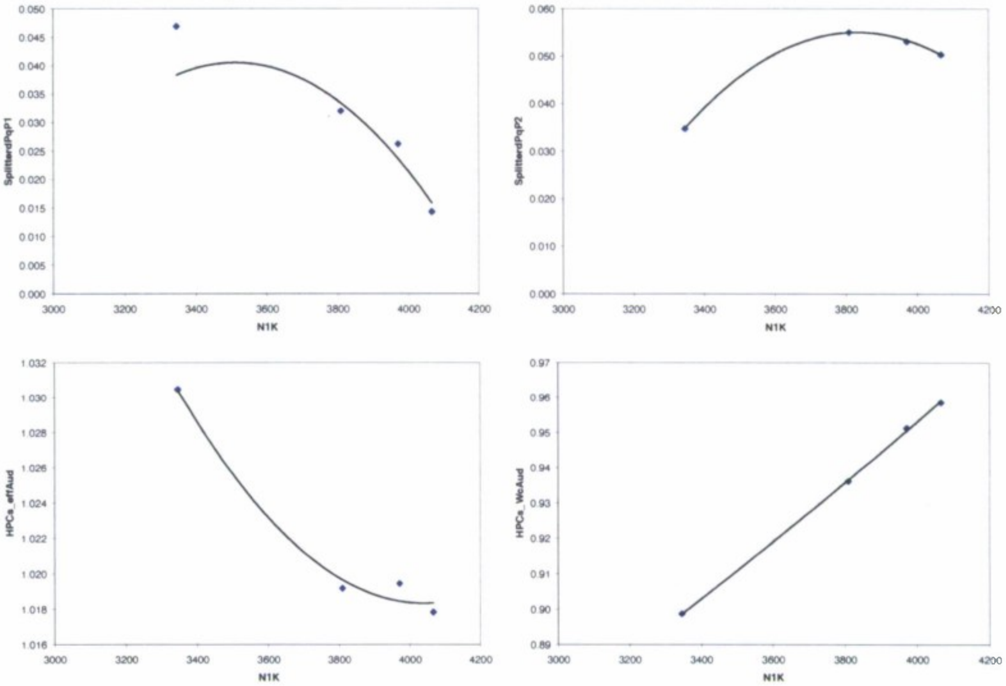


Figure 14: Modifier curve fits for Splitter and HPC

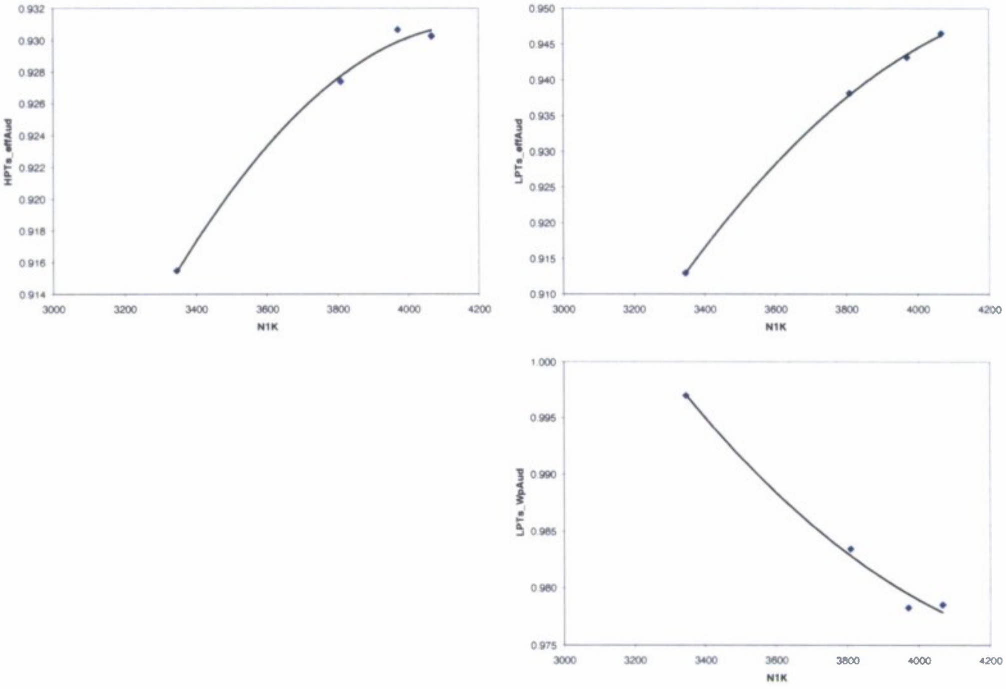


Figure 15: Modifier curve fits for HPT and LPT

As a final step in the demonstration, these curves were added back to the model, and a power hook was run to compare to the baseline and to the measured test data. Representative plots of fuel flow vs. EPR and thrust vs. fuel flow are shown below.

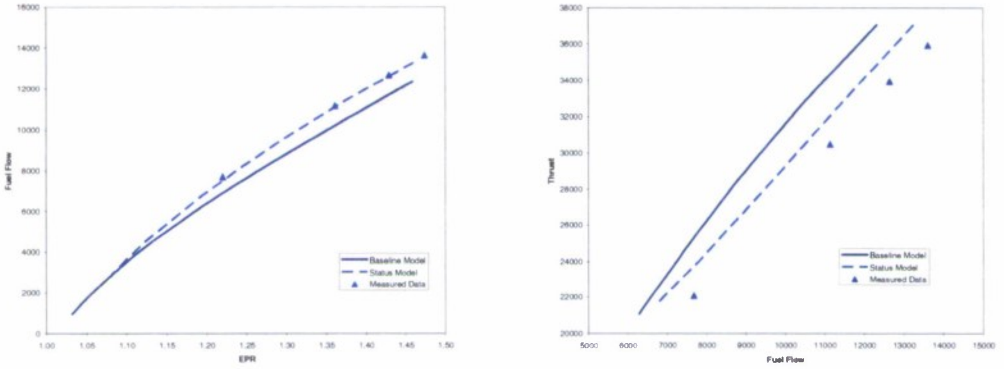


Figure 16: Representative Plots of Status Deck vs. Baseline and Test Data

It may be seen that the final results, while not perfect, are fairly good. Additional data and analyses could improve the results still further.

6.4.2 Entire Engine Population

The above results were obtained for a single representative engine chosen at random from the population of 46 engines. Once the status matching procedure was formulated using the SVD method with this single engine as a test case, the procedure was executed on the entire population of engines at four thrust bands each. This yielded 184 total points to perform the status match. This exercise was done in order to evaluate the overall speed and convergence properties of the algorithm when faced with large sets of engine data. The first attempt yielded a converged solution for 145 of the 184 test cases (79%). This result highlighted an important aspect of the SVD method; it relies on a local linearization of the model, so the final result is often dependent on the starting condition. A non-converged solution typically indicated an improper choice of starting condition for that particular data point. For this reason, some care must be taken when selecting a starting condition about which to begin linearization of the model. For this specific example, the poor convergence on the initial test was believed to be the result of the lower fidelity baseline model. However, when the data was rerun using a different starting condition, 94% convergence rate was achieved. In terms of computation time, all computations performed for matching of the baseline model to 184 different test points took approximately six minutes.

6.5 Results of FMC method

While the concept and implementation of a filtered Monte Carlo simulation is quite simple, the interpretation of its result is not. Typically, the interpretation involves various multivariate statistical analyses such as correlation analysis, principal component analysis, and Fisher's linear discriminant analysis. In this section it will be demonstrated how these techniques can be used to analyze a filtered Monte Carlo simulation result.

For the demonstration a filtered Monte Carlo simulation was performed on a set of measurements obtained from a real Pratt & Whitney PW2037 engine at a fixed operating condition. The range of each performance modifier is listed in Table 10. Although for an exploratory study it typically is desired to usc a wide range of values for each variable, for this particular case the ranges werc chosen based on the experience of the authors.

Table 10: Ranges of the Engine Performance Modifiers

Performance Scalar	Lower Bound	Upper Bound
Fan.s_effAud	0.98	1.01
Fan.s_WcAud	0.95	0.98
Duct4.s_dPqPaud	0.98	1.02
LPC.s_effAud	1.05	1.08
LPC.s_WcAud	0.9	0.93
Duct6.s_dPqPaud	0.98	1.02
HPC.s_effAud	1.01	1.04
HPC.s_WcAud	0.94	0.8
Burner.s_dPqPaud	0.99	1.02
HPT.s_WpAud	0.93	0.96
HPT.s_effAud	0.91	0.94
Duct11.s_dPqPaud	0.98	1.02
LPT.s_WpAud	0.96	0.99
LPT.s_effAud	0.93	0.96
Duct13.s_dPqPaud	0.98	1.01
Duct15.s_dPqPaud	0.98	1.02
Splitter.dPqP1	0	0.01
Splitter.dPqP2	0.045	0.055
Core_Nozz.s_CdTh	0.98	1.02
Byp_Nozz.s_CdTh	0.98	1.02

The target measurements used in this demonstration were FN, N1, N2, WF_AVG, PT25, PS31, TT25, TT35, TT49, and EGT. Precise information regarding the uncertainty of each measurement was unavailable to the authors so the filter tolerance was arbitrarily chosen as 1% of the target value. A total 160,000 simulations were performed, and 655 of these cases fell within the filter tolerance.

6.5.1 Histograms

The results of ultimate interest to the analyst are the values of the engine performance modifiers and, sometimes, the variances of the values. An estimate of an engine performance modifier and its variance can be graphically shown with histograms.

Figure 17 shows the histograms of five engine performance modifiers from the filtered Monte Carlo simulation: Fan.s_WcAud, LPC.s_WcAud, HPC.s_WcAud, HPT.s_WcAud, and LPT.s_effAud. The x-axis of each of the histograms is frequency, and the y-axis is the value of a modifier. In each plot it can be seen that some intervals are more frequently visited than others. The rest of the engine performance modifiers, not shown in Figure 17, produce nearly uniform histograms, which means that no particular value in the range of the modifier is preferred.

Should a point estimate of each modifier be required, the sample mean can be used as the point estimate. The samples means of the performance modifiers are listed in Table 11, and the NPSS output with the mean vector as an input is listed in Table 12. Each target variable is matched with less than 0.5% error. It should be noted that these results, when compared to the first column of Table 8 (the SVD results for the same data point) are similar but not identical. The SVD results are, however, well within the ranges shown in the histograms.

Table 11: Means of the Performance Modifiers

Performance Scalar	Mean
Fan.s_effAud	0.9970
Fan.s_WcAud	0.9660
Duct4.s_dPqPaud	1.0007
LPC.s_effAud	1.0653
LPC.s_WcAud	0.9220
Duct6.s_dPqPaud	1.0003
HPC.s_effAud	1.0251
HPC.s_WcAud	0.9622
Burner.s_dPqPaud	1.0045
HPT.s_WpAud	0.9475
HPT.s_effAud	0.9248
Duct11.s_dPqPaud	0.9998
LPT.s_WpAud	0.9746
LPT.s_effAud	0.9478
Duct13.s_dPqPaud	0.9950
Duct15.s_dPqPaud	0.9999
Splitter.dPqP1	0.0049
Splitter.dPqP2	0.0498
Core_Nozz.s_CdTh	1.0002
Byp_Nozz.s_CdTh	1.0002

Table 12: Matching Results

	NPSS Output	Measurement	% Error
FN	36040.80	36038.56	0.01
N1	3994.80	3993.86	0.02
N2	11351.40	11350.77	0.01
WF_AVG	13221.00	13232.23	-0.08
PT25	37.59	37.60	-0.02
PS31	373.17	373.30	-0.03
TT25	688.00	686.96	0.15
TT35	1403.20	1400.98	0.16
TT49	1470.20	1464.50	0.39
EGT	1470.17	1476.27	-0.41

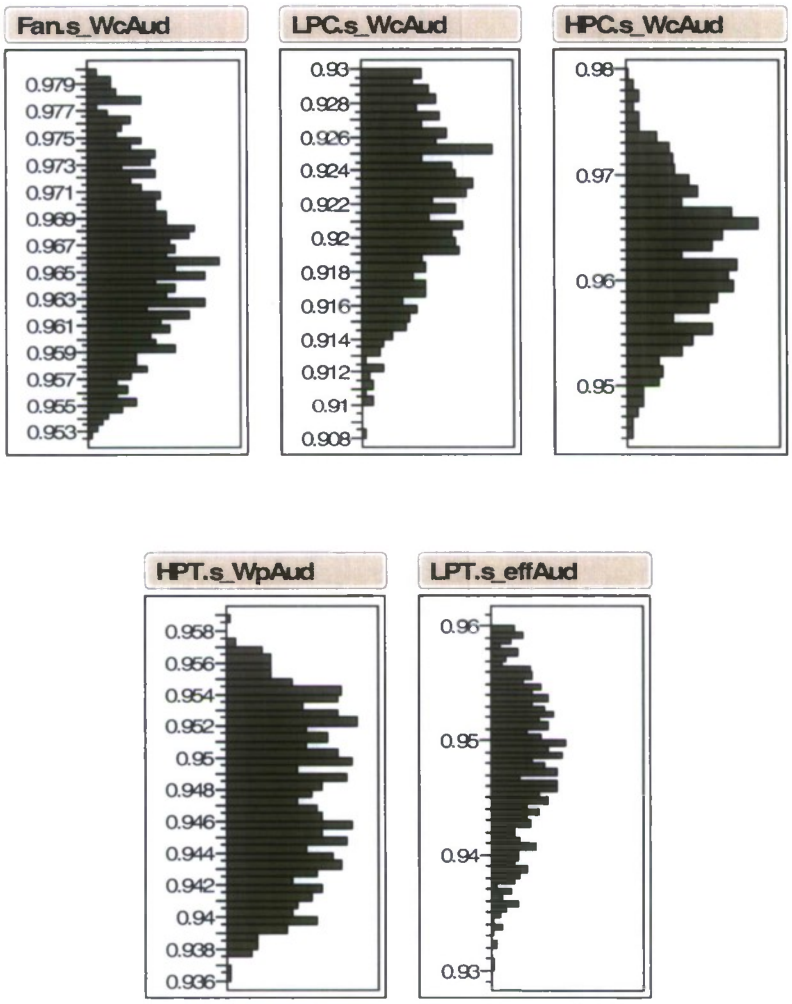


Figure 17: Non-uniform Histograms of Performance Modifiers

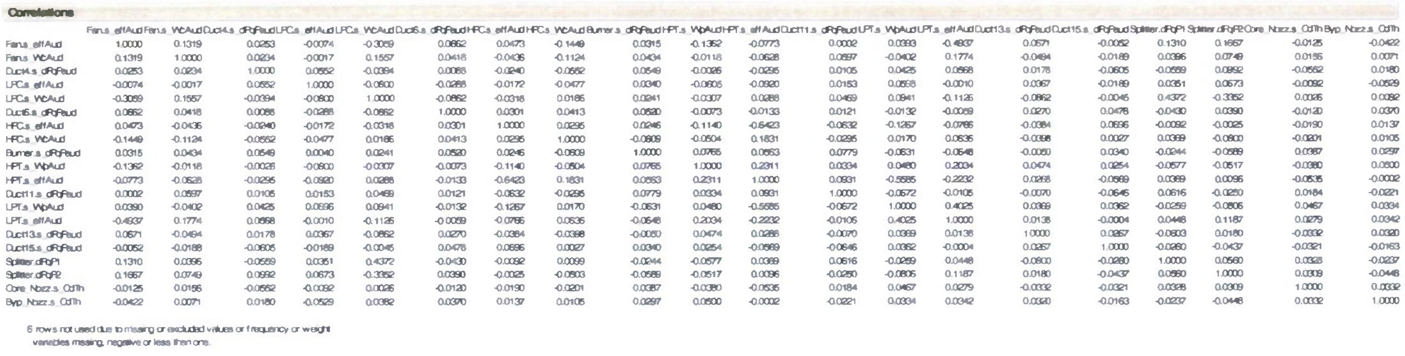
6.5.2 Correlations

Histograms provide useful information regarding univariate data. Multivariate data analysis of the correlation between two variables is also quite useful. A correlation coefficient quantitatively indicates the degree of correlation between two variables. The correlation coefficient is a real value between -1 and 1. The correlation coefficient of -1 or 1 means a perfect negative or positive correlation, respectfully. The correlation coefficient of zero means no correlation at all. Given a finite sample, a sample correlation coefficient between two variables *x* and *y* is defined by

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

where *n* is the number of samples [41]. Table 13 shows the matrix of the sample correlation coefficients. Among the pairs of 20 engine performance modifiers, three pairs are found more or less correlated, i.e., the absolute correlation coefficient equal or greater than 0.5. These three pairs are HPT efficiency modifier and LPT flow modifier, HPC efficiency modifier and HPT efficiency modifier, and fan efficiency modifier and LPT efficiency modifier. All of the three pairs are negatively correlated; when one increases, the other decreases.

Table 13: Correlation Coefficients



Correlations

	Fan.s_effAud	Fan.s_WcAud	Duct4.s_dPqPaud	LPC.s_effAud	LPC.s_WcAud	Duct6.s_dPqPaud	HPC.s_effAud	HPC.s_WcAud	Burner.s_dPqPaud	HPT.s_WpAud	HPT.s_effAud	Duct11.s_dPqPaud	LPT.s_WpAud	LPT.s_effAud	Duct13.s_dPqPaud	Duct15.s_dPqPaud	Splitter.dPqP1	Splitter.dPqP2	Core_Nozz.s_CdTh	Byp_Nozz.s_CdTh
Fan.s_effAud	1.0000	0.1319	0.0253	-0.0074	-0.3069	0.0662	0.0473	-0.1449	0.0315	-0.1362	-0.0773	0.0002	0.0390	-0.4937	0.0671	-0.0052	0.1310	0.1667	-0.0125	-0.0422
Fan.s_WcAud	0.1319	1.0000	0.0234	-0.0017	0.1557	0.0418	-0.0436	-0.1124	0.0434	-0.0118	-0.0628	0.0597	-0.0402	0.1774	-0.0494	-0.0189	0.0396	0.0749	0.0156	0.0071
Duct4.s_dPqPaud	0.0253	0.0234	1.0000	0.0552	-0.0394	0.0086	-0.0240	-0.0552	0.0549	-0.0026	-0.0296	0.0105	0.0425	0.0568	0.0178	-0.0605	-0.0559	0.0992	-0.0552	0.0180
LPC.s_effAud	-0.0074	-0.0017	0.0552	1.0000	-0.0600	-0.0288	-0.0172	-0.0477	0.0040	-0.0606	-0.0920	0.0153	0.0696	-0.0010	0.0367	-0.0189	0.0351	0.0673	-0.0092	-0.0529
LPC.s_WcAud	-0.3069	0.1557	-0.0394	-0.0600	1.0000	-0.0662	-0.0318	0.0186	0.0241	-0.0307	0.0288	0.0469	0.0941	-0.1126	-0.0862	-0.0046	0.4372	-0.3352	0.0026	0.0382
Duct6.s_dPqPaud	0.0662	0.0418	0.0086	-0.0288	-0.0662	1.0000	0.0301	0.0413	0.0520	-0.0073	-0.0133	0.0121	-0.0132	-0.0059	0.0270	0.0478	-0.0430	0.0390	-0.0120	0.0370
HPC.s_effAud	0.0473	-0.0436	-0.0240	-0.0172	-0.0318	0.0301	1.0000	0.0295	0.0246	-0.1140	-0.6423	-0.0632	-0.1267	-0.0786	-0.0384	0.0696	-0.0092	-0.0025	-0.0190	0.0137
HPC.s_WcAud	-0.1449	-0.1124	-0.0552	-0.0477	0.0186	0.0413	0.0295	1.0000	-0.0609	-0.0504	0.1831	-0.0296	0.0170	0.0635	-0.0398	0.0027	0.0369	-0.0803	-0.0201	0.0105
Burner.s_dPqPaud	0.0315	0.0434	0.0549	0.0040	0.0241	0.0520	0.0246	-0.0609	1.0000	0.0765	0.0653	0.0779	-0.0631	-0.0648	-0.0060	0.0340	-0.0244	-0.0589	0.0387	0.0297
HPT.s_WpAud	-0.1362	-0.0118	-0.0026	-0.0600	-0.0307	-0.0073	-0.1140	-0.0504	0.0765	1.0000	0.2311	0.0334	0.0480	0.2034	0.0474	0.0254	-0.0577	-0.0517	-0.0380	0.0600
HPT.s_effAud	-0.0773	-0.0628	-0.0296	-0.0920	0.0288	-0.0133	-0.6423	0.1831	0.0653	0.2311	1.0000	0.0931	-0.5585	-0.2232	0.0288	-0.0569	0.0369	0.0096	-0.0535	-0.0002
Duct11.s_dPqPaud	0.0002	0.0597	0.0105	0.0153	0.0469	0.0121	-0.0632	-0.0295	0.0779	0.0334	0.0931	1.0000	-0.0672	-0.0105	-0.0070	-0.0646	0.0616	-0.0250	0.0184	-0.0221
LPT.s_WpAud	0.0390	-0.0402	0.0425	0.0696	0.0941	-0.0132	-0.1267	0.0170	-0.0631	0.0480	-0.5585	-0.0672	1.0000	0.4025	0.0369	0.0362	-0.0259	-0.0806	0.0467	0.0334
LPT.s_effAud	-0.4937	0.1774	0.0568	-0.0010	-0.1126	-0.0059	-0.0786	0.0635	-0.0648	0.2034	-0.2232	-0.0106	0.4025	1.0000	0.0138	-0.0004	0.0448	0.1187	0.0279	0.0342
Duct13.s_dPqPaud	0.0671	-0.0494	0.0178	0.0367	-0.0862	0.0270	-0.0384	-0.0398	-0.0060	0.0474	0.0288	-0.0070	0.0369	0.0138	1.0000	0.0267	-0.0603	0.0180	-0.0332	0.0320
Duct15.s_dPqPaud	-0.0052	-0.0188	-0.0605	-0.0189	-0.0046	0.0478	0.0696	0.0027	0.0340	0.0254	-0.0569	-0.0646	0.0362	-0.0004	0.0267	1.0000	-0.0280	-0.0437	-0.0321	-0.0163
Splitter.dPqP1	0.1310	0.0396	-0.0559	0.0351	0.4372	-0.0430	-0.0092	0.0099	-0.0244	-0.0577	0.0369	0.0616	-0.0259	0.0448	-0.0603	-0.0280	1.0000	0.0560	0.0328	-0.0237
Splitter.dPqP2	0.1667	0.0749	0.0992	0.0673	-0.3352	0.0390	-0.0025	-0.0803	-0.0589	-0.0517	0.0096	-0.0250	-0.0806	0.1187	0.0180	-0.0437	0.0560	1.0000	0.0309	-0.0448
Core_Nozz.s_CdTh	-0.0125	0.0156	-0.0552	-0.0092	0.0026	-0.0120	-0.0190	-0.0201	0.0387	-0.0380	-0.0535	0.0184	0.0467	0.0279	-0.0332	-0.0321	0.0328	0.0309	1.0000	0.0032
Byp_Nozz.s_CdTh	-0.0422	0.0071	0.0180	-0.0529	0.0382	0.0370	0.0137	0.0105	0.0297	0.0600	-0.0002	-0.0221	0.0334	0.0342	0.0320	-0.0163	-0.0237	-0.0448	0.0032	1.0000

6 rows not used due to missing or excluded values or frequency or weight variables missing, negative or less than one.

A correlation coefficient is merely a measure of linearity between two variables. The correlation coefficient of zero does not mean they are not related to each other. If the two variables have a nonlinear relationship between them, their correlation coefficient is zero. Visualization of data helps avoid falling into this pitfall. Each case of a filtered Monte Carlo simulation is a multi-dimensional vector whose elements are independent variables and response variables. Correlation between a pair of independent variables can be visualized by plotting all the cases of the filtered Monte Carlo simulation in the space constituted by the pair of independent variables. If there is a correlation between this pair, the points of all the cases of the filtered Monte Carlo simulation will look alike more or less a band while those of uncorrelated pair build a cloud looking alike a circle or square

Figure 18 is the matrix of the correlation plots of seven variables, HPC efficiency modifier, HPC flow modifier, Burner pressure drop modifier, HPT flow modifier, HPT efficiency modifier, Duct pressure drop, and LPT flow modifier. Each cell of the matrix contains the correlation plot between two of the seven variables. The matrix of the correlation plots is symmetric; the lower diagonal is a flipped image of the upper one. According to the correlation analysis HPT efficiency modifier is more or less correlated with HPC efficiency modifier, HPT flow modifier, and LPT flow modifier. In this simulation none of the plots shows nonlinearity. Note that the observed correlations between two of the engine performance modifiers are valid only with the given measurements. They may or may not true at different operating conditions. When a severe correlation is present between two of engine performance modifiers, caution is required. Two or more sets of engine performance modifiers can result in an identical NPSS output.

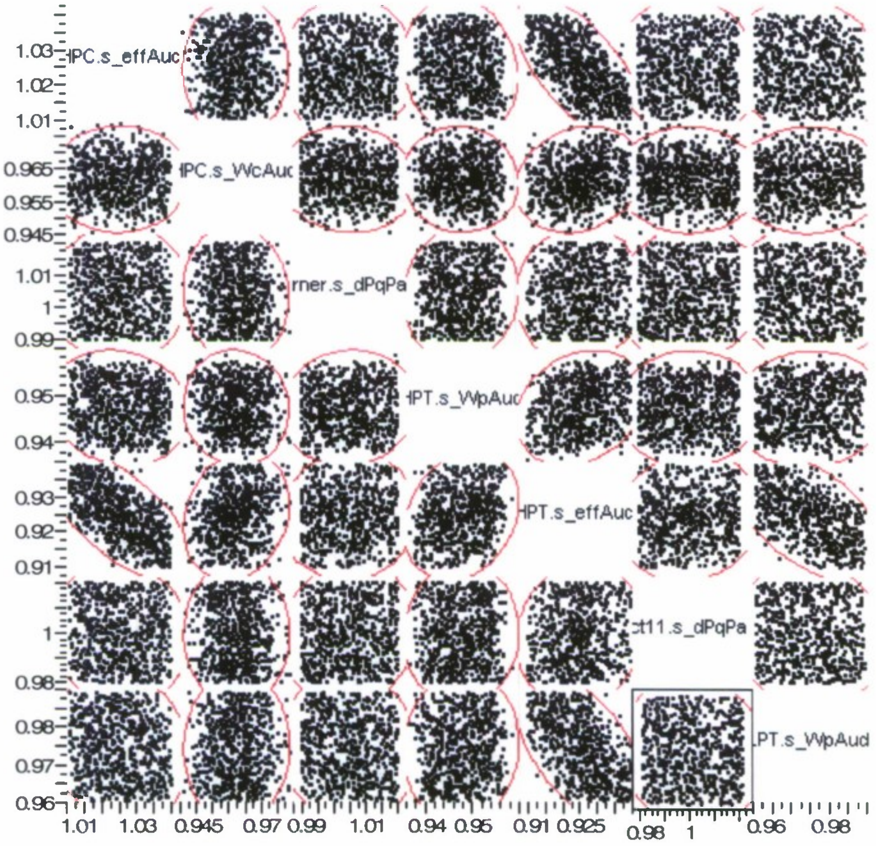


Figure 18: Correlation Plots

6.5.3 Principal Component Analysis

The correlation matrix and plots reveal the relationship between each pairs of variables. However, each case of the filtered Monte Carlo simulation is shown in several two dimensional space so that it is still hard to see how each case is distributed in the inverse solution space constituted by the independent variables. In fact, it is impossible due to the large dimensionality of the problem. The large dimensionality can be reduced principal component analysis (PCA).

Principal component analysis is a dimensionality reduction technique based on projection. Multi-dimensional data points are projected onto a few principal components, which constitute a lower dimensional space. The multidimensional data points can be visualized in the lower dimensional space, e.g., a 2-D or 3-D space. This kind of visualization is especially useful for identifying multiple solutions. Similar solutions will gather together and constitute a cloud of points while different solutions will appear apart from each other.

A principal component analysis follows the following steps [5]. Consider a data set of N observations $\{\mathbf{x}_n\}$ where $n = 1, \ldots, N$, and $\mathbf{x}_n$ is a vector with dimensionality D.

1. Subtract the mean from the data
The mean can be calculated with
$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$
The data subtracted by the mean vector is called the *adjusted* data:
$$\mathbf{x}_{n,adj} = \mathbf{x}_n - \overline{\mathbf{x}}$$

2. Calculate the covariance matrix
The sample covariance matrix can be calculated with
$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \overline{\mathbf{x}})(\mathbf{x}_n - \overline{\mathbf{x}})^T$$

3. Calculate the eigenvectors and eigenvalues of the covariance matrix
The principal components are the eigenvectors of the following problem:
$$\mathbf{Su} = \lambda\mathbf{u}$$
where **u** is a D-dimensional vector, and lambda is a modifier. The D values of lambda that satisfy the above equation are the eigenvalues, and the corresponding values of **u** are the eigenvectors. Refer Golub and van Loan [Golub and van Loan] for algorithms to find eigenvectors and eigenvalues.

4. Choose components
The eigenvector associated with the largest eigenvalue is called the first principal component. Typically a few principal components are important. When we project the data to these few principal components, we do not lose much of information even though the dimensionality of data decreases. For 2-D visualization two principal components should be chosen, and for 3-D visualization three principal components.

5. Project the data onto the components
The adjusted data can be projected on to the principal components using
$$\mathbf{x}_{proj} = \mathbf{F}^T\mathbf{x}_{adj}$$
where F is the matrix with the selected principal components in its columns.

Figure 19 shows all the cases of the filtered Monte Carlo simulation projected on to two the principal components. Note that the larger cloud of points on the right is the cases from the original ranges of variables and the smaller cloud on the left is the cases from another set of ranges. The separated clouds mean two different inverse solutions. By visualizing the multi-dimensional data in the two-dimensional space, the multiple solutions can be easily identified.

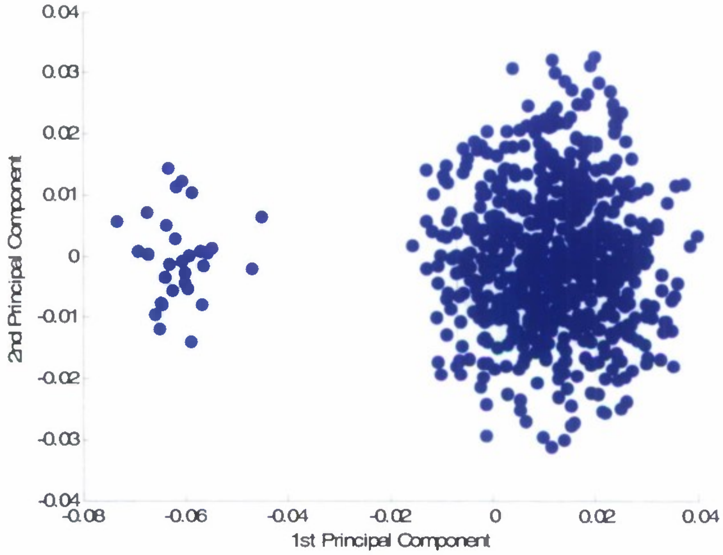


Figure 19: Projected Data onto the Principal Component Space

6.5.4 Fisher's Linear Discriminant Analysis (LDA)

PCA is the projection of the data onto a lower dimensional space such that the variance of the projected data is maximized [5]. Fisher's LDA is also a projection technique used for dimensionality reduction. However, unlike PCA, it is the projection of data that maximizes the class separation [5]. A Fisher's LDA follows the following steps [8]. Consider a data set of N observations $\{\mathbf{x}_n\}$ where $n = 1, \ldots, N$, and $\mathbf{x}_n$ is a vector with dimensionality D.

1. Calculate the within-scatter matrix

The within-scatter matrix for class c can be written as

$$\mathbf{S}_c = \sum_{\mathbf{x}\in\mathbf{x}_c}(\mathbf{x}-\overline{\mathbf{x}}_c)(\mathbf{x}-\overline{\mathbf{x}}_c)^T$$

And, then, the within-scatter matrix is

$$\mathbf{S}_w = \sum_{c\in C}\mathbf{S}_c$$

where C is the class space.

2. Calculate the between-scatter matrix

The between-scatter matrix can be calculated with

$$\mathbf{S}_b = \sum_{c\in C} n_c(\overline{\mathbf{x}}_c-\overline{\mathbf{x}})(\overline{\mathbf{x}}_c-\overline{\mathbf{x}})^T$$

where n_c is the number of data points in class c.

3. Calculate the optimal discriminant direction

The optimal discriminant direction φ can be obtained by solving the generalized eigenvalue problem:

$$\mathbf{S}_b\varphi = \lambda \mathbf{S}_w^{-1}\varphi$$

where λ is a scalar. The vector associated with the largest λ is the optimal discriminant direction.

Each element of the discriminant direction vector is the contribution of each variable in the original vector xn. He et al. [18] refer to the bar chart of the discriminant direction vector as the *contribution plot* and use it to determine which variables are responsible for the separation of classes.

Fisher's LDA is performed on all the cases of the filtered Monte Carlo simulation, and Figure 20 shows the contribution plot. Each bar represents the contribution of each performance modifier in distinguishing one cloud in Figure 19 from the other. Among 20 engine performance modifiers only one modifier, Splitter.dPqP2, is mainly separating the two clouds.

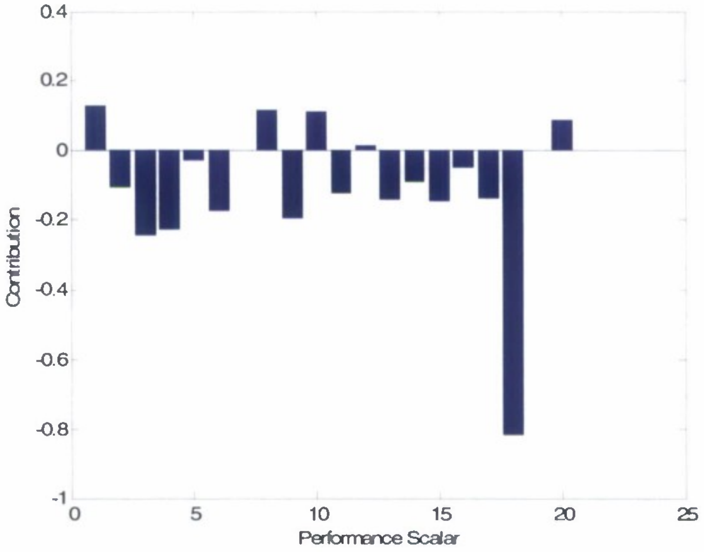


Figure 20: Contribution Plot

6.5.5 Additional Monte Carlo Studies

While the filtered Monte Carlo is a heuristic technique for optimization given mostly to trial and error, there are a number of plots that aid in the status matching and overall execution of the algorithm. These plots can provide insight into the filtering behavior as well as justification for selecting total case number and tolerances when using the method. This in turn gives additional scientific rigor to the algorithm.

The first plot shown in Figure 21 provides the user with information about the number of cases that will remain after filtering the data according to a certain tolerance. In other words, if the cases were removed that did not meet any of the target values within the given tolerance level, how many cases would remain?

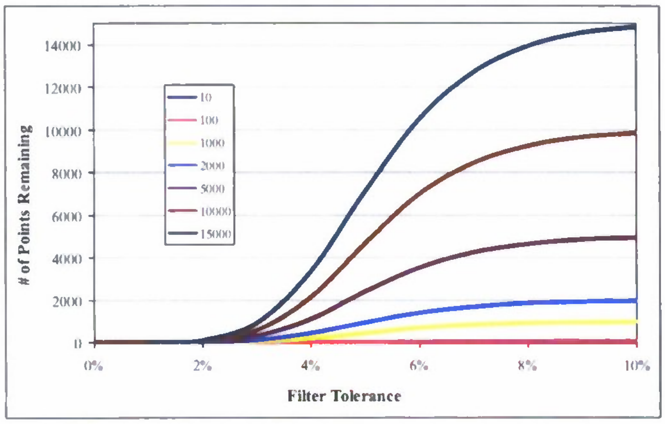


Figure 21: Number of Points Remaining for a Given Filter Tolerance

In the example in the above figure, if the simulation began with 10,000 cases and a four percent filter tolerance was placed on all of the targets, approximately 2000 cases would remain. This information is useful when determining what type of approach the user would like to take with the Monte Carlo method. If a probabilistic approach is desired, there may be a minimum number of final cases required to define frequency distributions or make statistical claims about the data.

The next figure indicates the minimum number of cases before the Monte Carlo simulation results become independent of that number.

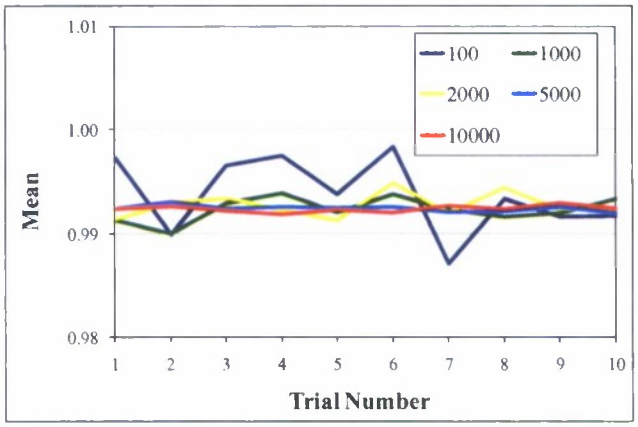


Figure 22: Monte Carlo Results Independence

For example, in the above figure, a simulation of 100 cases provides a different value of the mean each time the simulation is run. The mean line does not become horizontal until a simulation of 10,000 cases is run. Thus, to ensure accurate results and prove that a single simulation can be used to predict the modifier, a minimum of 10,000 cases is required.

7 Conclusions and Recommendations

7.1 Impact / Review of Significant Results

This research program developed an improved, automated process for calibrating turbine engine performance models. After a detailed examination of several potential algorithms, two complementary approaches were investigated: the Filtered Monte Carlo method and the Singular Value Decomposition method. The proposed algorithms meet the sponsor's requirements for a robust, fast process suitable for inexperienced users. Both methods were demonstrated to successfully match measured data with no prior knowledge of the engine. The procedure will contribute to Air Force goals for improved engine test planning, diagnostics, and condition-based maintenance.

One of the major conclusions of the research is that the choice of solution algorithm is not the most significant issue. All the algorithms investigated are theoretically similar to each other. For example, the Kalman filter is a time-dependent expansion of the weighted least squared method. SVD finds the minimum norm solution among possibly multiple solutions. Regularization methods use various norms such as L^1 and L^2 norms. The regularization term in the regularization methods work the same as prior information in Bayesian methods. Genetic algorithms are a version of the filtered Monte Carlo method with a heuristics inspired by natural selection. The main difference between these algorithms is the amount of information required to run the algorithms. Bayesian methods require the most amount of prior information while the filtered Monte Carlo method the least.

The most significant factor in a successful method is the user's choice of modifiers. Even a sophisticated Bayesian method may result in erroneous solutions if it includes unnecessary modifiers. It is difficult to avoid the fact that choosing the right set of modifiers requires experience; although certain steps in the process have been automated and effectively prompt the user when a decision is required. It is in this respect that the FMC and SVD methods complement each other in a powerful way. An initial FMC analysis will identify for an inexperienced user which variables are significant and will provide him or her with appropriate ranges for those variables. This analysis may need to be conducted only one time, at the beginning of the test program. The SVD method may then be used to quickly determine the best value for each modifier at all subsequent test conditions.

It is noteworthy that, if user experience can be translated into probability distributions for prior assumptions on the modifiers, then Bayesian methods can easily be incorporated into the developed process. Over the long term, this may prove to be the best approach to the problem.

7.2 Transition / Collaboration Opportunities

The engine status matching process is applicable to calibration of other types of models. Similar methods have been used by the researchers for calibration of engine, aircraft, noise, and emissions models and for calibration of lower fidelity aero models to CFD models.

7.3 Recommendations for Future Work

Recommendations for future work address to the two main obstacles to completely automating the status matching process, which are (1) selecting the most appropriate set of modifiers and (2) regressing the results.

7.3.1 Modifier selection

The first step in the creation of a status deck is to select which component model modifiers will be allowed to vary to force the model to match the test-measured data. As in all model fitting problems, it is important to avoid Type I errors (i.e., including unnecessary variables) as well as Type II errors (i.e., omitting significant variables) while being alert to multicollinearity problems. No automated process can completely replace the expertise of the analyst. However, it has been demonstrated in the present work that the filtered Monte Carlo method provides significant insight which can guide an inexperienced analyst.

Unfortunately, the filtered Monte Carlo procedure can be quite time consuming. Future research should focus on methods which are faster. One approach that has been recently investigated [25] involves a Bayesian approach where the analyst may assign prior distributions to the modifiers. Through an iterative refinement procedure, the most effective explanatory variables congruent with prior expectations may be identified. It is recommended that research continue into applications of this approach to the status matching problem.

7.3.2 Regression

The final step in the creation of a status deck is to regress the modifiers against physical parameters such as corrected rotor speeds, and incorporating these curves into the engine model. In the present research this was a manual process. The process requires additional expertise on the part of the analyst, who must determine not only the proper modifiers to include in the matching step, but must also determine the most appropriate parameters against which to regress those modifiers. The process is especially difficult and time consuming when analyzing data from multiple flight conditions. In the demonstration problem considered in the present research, all the data was taken at sea level static conditions. At this flight condition, the modifiers were found to be either constant values or simple functions of the rotor speeds. However, data taken at high altitude, low Mach number conditions might be expected to be a function of Reynolds number, and data taken at high Mach conditions might be expected to be a function of turbomachinery running clearances. These and other parameters must be considered in the regression process.

Ideally, it would be possible to match and regress data from multiple flight conditions and multiple power settings all in one step. Some attempts at this have been reported in the literature for related problems (see for example [29]). Typical approaches require the assumption of some functional form, e.g. a second degree polynomial. It may be appreciated even from the limited analysis performed on the sea level data in the present work that a simple polynomial is not sufficient. Clearly more research is required in this area.

8 Acknowledgements

This work was performed under a grant from the Air Force Office of Scientific Research. The authors would like to thank Dr. J.D. Schmisseur and Lt. Col. Rhett Jefferies from AFOSR, and Messrs. Rob McAmis and Mark Chappell from Arnold Engineering Development Center for their guidance and support. The authors would also like to thank Delta Air Lines for providing the engine data.

9 References

[1] *Aircraft Engine Emissions*, <http://www.caa.co.uk/default.aspx?catid=702> (July 21, 2008), Civil Aviation Authority.

[2] Bertero, M., "Linear Inverse and Ill-Posed Problems," *Advances in Electronics and Electron Physics*, Vol. 75, pp. 1-120, 1989.

[3] Bertero, M. and Boccacci, P., *Introduction to Inverse Problems in Imaging*, Taylor & Francis, 1998.

[4] Beven, K.J., Binley, A.M., "The future of distributed models: model calibration and uncertainty prediction", *Hydrological Processes*, Vol. 6, pp. 279–298, 1992.

[5] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, 2006.

[6] Björk, Å., *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.

[7] Breese, J. S., Horvitz, E. J., Peot, M. A., Gay, R., and Quentin, G. H., "Automated Decision-Analytic Diagnosis of Thermal Performance in Gas Turbines," in 1992 ASME Turbo Expo, Cologne, Germany, June 1992.

[8] Chiang, L. H., Russell, E. L., and Braatz, R. D., "Fault Diagnosis in Chemical Processes Using Fisher Discriminant Analysis, Discriminant Partial Least Squares, and Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, Vol. 50, pp. 243-252, 2000.

[9] Dieter, George, *Engineering Design: A Materials and Processing Approach, 3rd Edition*, McGraw Hill, 2000.

[10] Doel, D. L., "TEMPER - A Gas Path Analysis Tool for Commercial Jet Engines," *Journal of Engineering for Gas Turbines and Power*, Vol. 116, pp. 82-89, 1994.

[11] Elkan, C., "The Paradoxical Success of Fuzzy Logic," in Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), Washington, D. C., pp. 698-703, MIT Press, July 1993.

[12] Ganguli, R., "Application of Fuzzy Logic for Fault lsolation of Jet Engines," *Journal of Engineering for Gas Turbines and Power*, Vol. 125, pp. 617-623, July 2003.

[13] Grimm, L.G. and Yarnold, P.R., eds., *Reading and Understanding Multivariate Statistics*, American Psychological Association, Washington, D.C., 1994.

[14] Golub, G. H. and van Loan, C. F., *Matrix Computations*, Third ed., Johns Hopkins University Press, 1996.

[15] Habrard, Alain G., "Characterization of Components Performance and Optimization of Matching in Jet Engine Development", in *Modern Prediction Methods for Turbomachine Performance*, AGARD LS-83, NATO Advisory Group for Aerospace Research and Development, 1976.

[16] Hadamard, J., *Lectures on Cauchy's Problem in Linear Partial Differential Equations*, Dover, 1952.

[17] Hansen, P. C., *Rank-Deficient and Discrete Ill-Posed Problems*, Society for Industrial and Applied Mathematics (SIAM), 1998.

[18] He, Q. P., Qin, S. J., and Wang, J., "A New Fault Diagnosis Method Using Fault Directions in Fisher Discriminant Analysis," *AIChe Journal*, Vol. 51, No. 2, February 2005.

[19] Julier, S. J. and Uhlmann, J. K., "A New Extension of the Kalman Filter to Nonlinear Systems," in the SPIE AeroSense Symposium, April 21–24, 1997, Orlando, FL, USA.

[20] Kaipio, J. and Somersalo, E., *Statistical and Computational Inverse Problems*, Springer, 2004.

[21] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, pp. 35-45, 1960.

[22] Keller, J. B., "Inverse Problems," *The American Mathematical Monthly*, Vol. 83, pp. 107-118, 1976.

[23] Kennedy, M. C., and O'Hagan, A., "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society Series B (Methodological)*, Vol. 63, No. 3, pp. 425–464, 2001.

[24] Kobayashi, T. and Simon, D. L., "Application of a Bank of Kalman Filters for Aircraft Engine Fault Diagnostics," in Proceedings of ASME Turbo Expo 2003, June 16-19, 2003, Atlanta, GA, USA.

[25] Lee, Y. K., *A Fault Diagnosis Technique for Complex Systems Using Bayesian Data Analysis*, Ph.D. Thesis, Georgia Institute of Technology, April 2008.

[26] Li, Y.G., "Performance-Analysis-Based Gas Turbine Diagnostics: A Review", *Proceedings of the Institute of Mechanical Engineering, Part A: Journal of Power and Energy*, Vol. 216, No. A5, 2002.

[27] Lorenz, E., "Empirical orthogonal functions and statistical weather prediction", Tech. Rep. 1, Statistical Forecasting Project, Department of Meteorology, Massachusetts Institute of Technology, Cambridge, MA, 1956.

[28] Mast, T. A., Reed, A. T., Yurkovich, S., Ashby, M., and Adibhatla, S., "Bayesian Belief Networks for Fault Identification in Aircraft Gas Turbine Engines," in Proceedings of the 1999 IEEE International Conference on Control Applications, Kohala Coast-Island of Hawai'l, Hawai'l, USA, August 22-27, 1999.

[29] Monaco, J., Kidman, D., Malloy, D., Ward, D.G., and Gist, J.F., "Automated Methods to Calibrate a High-Fidelity Thrust Deck to Aid Aeropropulsion Test and Evaluation," ASME Paper GT2008-50213, ASME Turbo Expo 2008, Berlin, Germany, June 9-13, 2008.

[30] Provost, M. J., "COMPASS: A Generalized Ground-Based Monitoring System," AGARD Conference Proceedings No. 448, 1988.

[31] Romessis, C. and Mathioudakis, K., "Bayesian Network Approach for Gas Path Fault Diagnosis," *Journal of Engineering for Gas Turbines and Power*, Vol. 128, pp. 65-72, January 2006.

[32] Simon, D. and Simon, D. L., "Constrained Kalman Filtering Via Density Function Truncation for Turbofan Engine Health Estimation," NASA TM-2006-214129, 2006.

[33] Sorenson, H. W., *Kalman Filtering: Theory and Application*, IEEE Press, 1985.

[34] Tarantola, A., *Inverse Problem Theory and Methods for Model Parameter Estimation*, Society for Industrial and Applied Mathematics, 2004.

[35] Tibshirani, R., "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 58, No. 1, 1996, pp. 267-288.

[36] Tikhonov, A. N., Goncharsky, A. V., and Stepanov, V. V., *Numerical Methods for the Solution of Ill-Posed Problems*, Kluwer Academic Publishers, 1995.

[37] Torella, G. and Federico, M., "The Optimization of Utilization of Gas Turbine Engines for Aircraft Propulsion," in the 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, June 20-24, 1999, Los Angeles, CA, USA.

[38] Urban, L., "Gas Path Analysis Applied to Turbine Engine Condition Monitoring," in the 8th AIAA and ASME Joint Propulsion Specialist Conference, New Orleans, Louisiana, November 29- December 1, 1972.

[39] Vanderplaats, Garret, *Numerical Optimization Techniques for Engineering Design 4th Edition*, Vanderplaats Research and Development, Inc., Colorado Springs, 2005.

[40] Watkins, D. S., *Fundamentals of Matrix Computations*, Wiley-Interscience, 2002.

[41] Weisstein, E. W., "Correlation Coefficient," From MathWorld-A Wolfram Web Resource, http://mathworld.wolfram.com/CorrelationCoefficient.html, Accessed 07-October-2008.

[42] Zedda, M. and Singh, R., "Fault Diagnosis of a Turbofan Engine Using Neural Networks: A Quantitative Approach," AIAA 98-3602, 1998.

[43] Zedda, M. and Singh, R., "Gas Turbine Engine and Sensor Fault Diagnosis Using Optimization Techniques," in the 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, July 1999, Cleveland, OH, USA.